(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

CORRECTED VERSION

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 September 2001 (13.09.2001)

PCT

(10) International Publication Number
WO 01/67694 A1

(51) International Patent Classification[7]: H04L 12/56, H04Q 11/04

(21) International Application Number: PCT/US01/01003

(22) International Filing Date: 11 January 2001 (11.01.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/518,575 3 March 2000 (03.03.2000) US
09/518,526 4 March 2000 (04.03.2000) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:
US 09/518,526 (CON)
Filed on 4 March 2000 (04.03.2000)

(71) Applicant *(for all designated States except US)*: **CELOX NETWORKS, INC.** [US/US]; 1 Cabot Road, Hudson, MA 01749 (US).

(72) Inventors; and

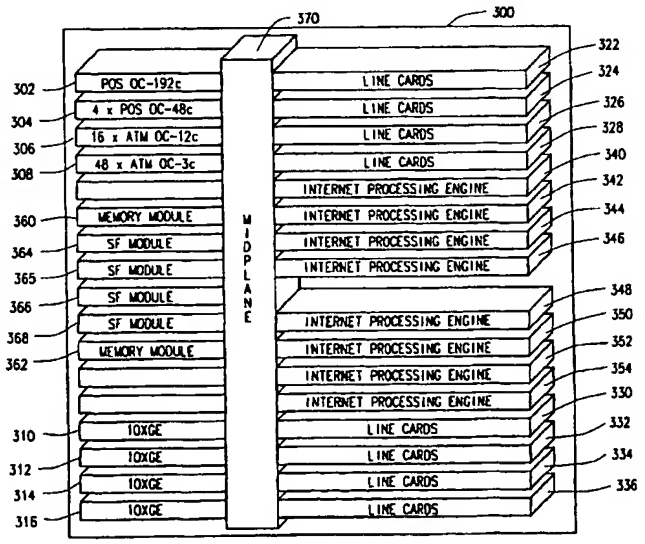
(75) Inventors/Applicants *(for US only)*: **BORDES, Jean, Pierre** [US/US]; 1109 Babler Forest Court, Chesterfield, MO 63005 (US). **SCHMID, Otto, Andreas** [DE/US]; 6625 Clayton Avenue #228, St. Louis, MO 63139 (US). **DAVIS, Curtis** [US/US]; 341 N. McKnight #D, St. Louis, MO 63108 (US). **MAHER, Monier** [DE/US]; 407 N. Taylor #302, St. Louis, MO 63108 (US). **HEGDE, Manju** [IN/US]; 10373 Gosport #4, St. Louis, MO 63146 (US).

(74) Agents: **HAFERKAMP, Richard, E.** et al.; Suite 1400, 7733 Forsyth Blvd., St. Louis, MO 63105-1817 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

[Continued on next page]

(54) Title: BROADBAND MID-NETWORK SERVER



(57) Abstract: A broadband mid-network server provides high-speed, reliable, secure, flexible, high-bandwidth, and easily managed access to the Internet to accommodate all current Internet services including email, file transfer, web surfing and e-commerce, as well as new value added services such as VoIP and Real Time Video. The preferred server is scalable in terms of both bandwidth and processing power. The server includes the ability to distribute traffic across a number of Internet processing engines and, more specifically, across a number of protocol processing units provided in each engine (the bandwidth to which can be coordinated), to provide compute power and state space required for performing per user processing for a large number of users.

WO 01/67694 A1

(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(48) **Date of publication of this corrected version:**
10 January 2002

(15) **Information about Correction:**
see PCT Gazette No. 02/2002 of 10 January 2002, Section II

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

BROADBAND MID-NETWORK SERVER

The present invention relates to internetworked communication systems, and especially (but not exclusively) to a highly scalable broadband mid-network server for performing mid-network processing functions including routing functions, per user processing, encryption, bandwidth distribution and traffic shaping.

Background and Summary of the Invention

As bandwidths within the core of the Internet increase, there is an increasing trend towards using the Internet Protocol ("IP") as the core network layer protocol for all kinds of traffic, including voice, video and data.

Historically, quality of service on the Internet has been what is called "best effort." That is, the network attempts to transport as much traffic as possible, but if there is insufficient capacity to handle the traffic, all connections are equally likely to be influenced by congestion. Thus, "best effort" implies that the Internet provides only one class of service to any connection, and that all connections are handled equally with no priority.

In the case of traditional Internet applications, this approach was often sufficient. However, the intrinsic potential of the Internet is considerably greater, and includes new multimedia and interactive applications. Voice over IP ("VoIP") and Real Time Video are envisioned to be two significant applications for propelling Internet growth to the next level. VoIP can be defined as the ability to make telephone calls and send faxes over IP networks. The benefits of this technology are cost reduction, simplification, consolidation and advanced applications such as shared screens or whiteboarding which combine voice and data. Real Time Video is a "direct-to-user" technique in which a video signal is transmitted to the client device and presentation of the video begins after a short delay for data buffering, and eliminates the need for significant client-site storage capacity. It is also expected to become popular with businesses. Related to this is webconferencing, which requires high bandwidth since it is a continuous transfer of image information together with voice transfer. Webconferencing also requires real time traffic handling because it is usually implemented as an interactive application.

All of these new applications will generally require significant bandwidth and/or reduced latencies. Bandwidth is the critical factor when large amounts of information must be transferred within a reasonable time period. Latency is the minimum time elapsed between requesting and receiving data and is important in real-time and interactive applications such as webconferencing and telecommuting. Presently, most telecommuters depend upon analog modems with limited bandwidth and significant latency for dial-up connectivity. Even for today's applications, dialup connectivity is often inadequate.

There are competing "last mile" technologies today which provide transport services to the user for delivering packets to the "edge" of the Internet. To complete the communication, the packets need to be formatted to allow them to enter the

Internet cloud and find their way to their respective destinations. The emergence of supporting protocols for new applications and the growth spurt in number of users and the required bandwidth per user results in a very dynamic access environment.

The following is a summary of observations that pertain to an ideal mid-network point within the Internet:

- In order to accommodate a variety of source packets, all the requisite protocols must be efficiently supported.
- Virtual Private Network services allow a private network to be configured within a public network. This is one of the drivers for Internet access amongst businesses. To allow Virtual Private Networks to coexist on the public Internet, and to encourage business use of the Internet, great care must be taken with respect to security and authentication issues, and tunneling protocols such as L2TP and IPSec must be efficiently supported.
- The number of subscribers handled by one system and the different qualities of service provided will make service provider administration more complex. To make provisioning of broadband access more attractive to service providers, subscriber management and usage accounting must be simplified, and differentiated services must be provided.
- Broadband makes it possible to provide different amounts of bandwidth to users and to smaller Internet Service Providers. To make wholesaling of IP connectivity possible, and to simplify service and repair functions, the ability to support multiple service providers with one mid-network server must be provided.
- A large number of connections are serviced with a broadband mid-network server. In order to ensure that service is not interrupted, the broadband server must

have very high availability. Such availability is also required for mission-critical business applications.

- Central office co-location space is limited. To conserve this space, large connection densities must be provided.
- When subscribers are allowed access at high speeds, it is possible for a limited number of users demanding disproportionate amounts of bandwidth to disrupt service for other customers. To ensure that large traffic bursts do not overload small client buffers, and to ensure that service providers and customers are treated fairly, traffic shaping must be provided.
- To enable new value-added services, large bandwidths and low latencies are critical.

In order to solve these and other needs in the art, the inventors hereof have succeeded at designing and developing a broadband mid-network server that, in the most preferred embodiment, satisfies all of the requirements described above. This inventive server provides reliable, secure, fast, flexible, high-bandwidth, and easily managed access to the Internet so as to accommodate all current Internet services including email, file transfer, web surfing and e-commerce, as well as the new value added services such as VoIP and Real Time Video. To meet these requirements, the broadband mid-network server of the present invention has been designed to scale not only in bandwidth, but also in processing power and state space. In the preferred embodiment, the architecture allows a service provider to configure the cards chosen for use in the available chassis space to suit his particular application. For example, to maximize processing power, a service provider could increase the number of IPE cards at the expense of a fewer number of line cards; as few as one line card. In the case of one line card, the maximum amount of processing power would be available to a service provider. IN the preferred embodiment described in detail below, this

configuration would provide 240 processors and 39 gigabytes of memory. This would allow for a greater number and complexity of value added services which require more processing power. Alternatively, a greater number of line cards could be selected for use in a chassis which would be desirable to handle greater traffic and throughput at the expense of fewer value added services.

The high bandwidth core routers that are currently under development by third parties are optimized for performing large numbers of fast routing lookups, but are not expected to provide generalized and flexible computing power for supporting the substantial amount of processing needed for, among other things, per user and per packet processing. In contrast, the broadband mid-network server of the present invention includes the ability to distribute traffic across a number of Internet processing engines and, more specifically, across a number of protocol processing units provided in each engine (the bandwidth to which can be coordinated), to provide compute power and state space required for performing per user processing for a large number of users.

One important feature of the present invention is a unique architectural philosophy, which provides that processing be performed as close to the physical layer as warranted by considerations of flexibility, cost and complexity. This architectural philosophy maintains balance between two kinds of processing which are important to scaling bandwidth with value-added services in broadband networks: time-consuming, repetitive processing; and flexible processing which must be easy to program by third parties. The need for considerable time-consuming repetitive processing, which has proved to create a bottleneck in the processor-based servers of the prior art, is addressed by the inventive architecture through specialized hardware, and results in dramatic increases in speed and decreases in delay. The need for

flexible, easy to use, computing power to enable service providers to scale with value-added services is addressed by the inventive architecture preferably through the provision of high-performance general purpose processors which are paralleled and which can be scaled to a virtually limitless degree. Alternatively, network processors or digital signal processors or any other programmable processor could be utilized as well. Accordingly, the broadband mid-network server of the present invention provides a system that is currently unrivalled in performance and which can become the prime mover of Internet services such as managed, secure VPNs, Voice over IP and Real Time Video.

While some of the principal features and advantages of the present invention have been described above, a greater and more thorough appreciation of the invention may be attained by referring to the drawings and the detailed description of the preferred embodiments which follow.

Brief Description of the Drawings

Fig. 1 illustrates a single shelf broadband mid-network server according to one embodiment of the present invention;

Fig. 2 is a functional block diagram of the preferred server shown in Fig. 1;

Fig. 3 is a functional block diagram of an exemplary line card shown in Figs. 1 and 2;

Fig. 4 is a functional block diagram of an exemplary IPE card shown in Figs. 1 and 2;

Fig. 5 illustrates routed distribution to an IPE card;

Fig. 6 illustrates the processing flow on an IPE card;

Fig. 7 illustrates a protocol processing platform according to the present invention;

Fig. 8 is a functional block diagram of an exemplary buffer access controller;

Fig. 9 illustrates the format of a cell received at an input to a BAC from a PIC;

Fig. 10 is a functional block diagram of a preferred packet manager;

Fig. 11 is an illustration of the deployment of a broad-band mid-network server at a Service Provider POP;

Fig. 12 is an illustraion of the different kinds of links an ISP may want on a secure segment;

Fig. 13 is an illustration of the system wide bandwidth distribution functions;

Fig. 14 is an illustration of the multi-level policing and multi-level shaping that occurs in the system;

Fig. 15 is an illustration of router distribution, two level policing, routing and two level shaping;

Fig. 16 is a functional block diagram of a preferred packet inspector;

Fig. 17 is an illustration of the preferred Distributor Flow Unit; and

Fig. 18 is a summary of the highlights of the DFU.

Detailed Description of the Preferred Embodiments

The mid-network processor of the present invention is preferably implemented in a single shelf system as shown generally in Fig. 1, and is indicated generally by reference character 300. As shown in Fig. 1, the mid-network processor 300 is provided with a number of physical connection ("PHY") cards 302-316 through which packets may enter and exit the mid-network processor 300 according to a particular communication protocol, as is

known in the art. For the preferred embodiment illustrated in Fig. 1, the mid-network processor 300 supports the POS, ATM, and Gigabit Ethernet layer two protocols, although the mid-network processor may readily be configured to support additional protocols, as will be apparent. The PHY cards 302-316 are each associated with line cards 322-336, respectively, as shown in Fig. 1. As is well known in the art, each PHY card is media specific. In other words, each PHY card is provided with connectors and other components necessary to interface with the communication media connected thereto, and over which packets enter and exit the PHY card. Each line card is configured to process packets of the type received from its associated PHY card, as explained more fully below.

The preferred mid-network processor 300 shown in Fig. 1 is also provided with a number of Internet Processing Engine ("IPE") cards 340-354, as well as two flash memory modules 360, 362 and four switch fabric modules 364-368. As appreciated by those skilled in the art, the number of switch fabric cards required is a function of the switch fabric card design as well as the desired redundancy overall performance. Fig. 1 also illustrates a midplane 370 that is provided for interconnecting the various cards described above. The preferred mid-network processor 300 utilizes a card-based approach to facilitate maintenance and expansion of the mid-network processor 300, as necessary, but this is clearly not a limitation of the present invention.

The manner in which packets are processed by the preferred mid-network processor 300 will now be described with reference to Fig. 2, which is a functional block diagram of the preferred mid-network processor 300 shown

in Fig. 1 (although, to simplify the illustration, Fig. 2 does not show the PHY cards 310-316, the line cards 330-336 and the IPE cards 346-354 shown in Fig. 1). Packets enter the mid-network processor 300 via the PHY cards, as is known in the art. Each PHY card then delivers its packets to its associated line card through the midplane 370. After performing initial processing of the packet, the line card delivers the packet again through the midplane to the switch fabric which, in turn, delivers the packet to one of the IPE cards for performing certain mid-network processing functions, such as routing functions, per user processing, encryption, and bandwidth distribution. After performing mid-network processing for the packet delivered thereto, the IPE card sends the packet back into the switch fabric, typically for delivery to one of the line cards for some additional processing before allowing the packet to exit the mid-network processor 300 through one of the PHY cards. In some cases, depending upon how the mid-network processor of the present invention is implemented, a single IPE card may be insufficient to complete the necessary mid-network processing functions for a packet delivered thereto. In this case, upon performing some processing, the IPE card will deliver the packet to another IPE card (rather than to one of the line cards) via the switch fabric for further processing. Thus, although a packet will typically be processed by only one IPE card, it is possible to process a packet in multiple IPE cards, if necessary.

In this preferred embodiment, all of the line cards contain identical hardware, but are independently programmable. Likewise, all of the IPE cards contain identical hardware, but are independently programmable.

This contributes to the scalability and elegantly simple design of the preferred mid-network processor 300. Additional processing power can be provided to the mid-network processor by simply adding additional IPE cards. Similarly, additional users can be supported by the mid-network processor 300 by adding additional line cards and PHY cards, and perhaps additional IPE cards to provide additional processing for the newly added users, if necessary.

The flash memory cards are provided for storing configuration data used by the IPE cards during system initialization.

Note that, as used herein, the term "packet" refers to any type of packet that enters or exits the mid-network processor 300, including packets input to the mid-network processor 300 in the form of cells (such as ATM cells) via an interleaved or non-interleaved cell stream.

In general, each line card used in the preferred mid-network processor 300 performs a number of functions. Initially, the line card converts packets (possibly of varying lengths) delivered thereto into fixed length cells. In this preferred embodiment, each line card converts input packets (including packets represented by individual cells) into 64 byte cells. The line card then examines the stream of fixed length cells "on the fly" to obtain important control information, including the protocol encapsulation sequence for each packet and those portions of the packet which should be captured for processing. This control information is then used on the line card to reassemble the packet, and to format the reassembled packet into one of a limited number of protocol types that are supported by the IPE cards.

Thus, while any given line card can be configured to support packets having a number of protocol layers and protocol encapsulation sequences, the line card is configured to convert these packets into generally non-encapsulated packets (or, stated another way, into packets having an encapsulation sequence of one) of a type that is supported by each of the IPE cards. The line card then sends the reassembled and formatted packet into the switch fabric (in the form of contiguous fixed length cells) for delivery to one of the IPE cards that was designated by the line card for further processing that particular packet.

Although the fixed length cells which comprise a packet are arranged back to back when the packet is delivered to the switch fabric by a line card, the cells may become interleaved with other cells destined for the same IPE card during the course of traversing the switch fabric. As a result, the cell stream provided by the switch fabric to any given IPE card may be an interleaved cell stream. Thus, the IPE card will first examine this cell stream "on the fly" (much like the cell stream examination conducted by the line cards, explained above) to ascertain important control information. The IPE card then processes this control information to perform routing look-ups and other mid-network processing functions for each packet delivered thereto. The control information is also used by the IPE card to reassemble each packet, and to format each packet according to the packet's destination interface. The IPE card then sends the reassembled and formatted packet back into the switch fabric in the form of contiguous fixed length cells for delivery to one of the line cards (or for delivery to another IPE card, in the case where additional mid-

network processing functions must be performed for the packet in question).

As noted above, although the cells of any given packet may enter the switch fabric in a back to back arrangement, these cells may become interleaved with other cells during the course of traversing the switch fabric. Thus, the stream of cells provided by the switch fabric to any given line card may be an interleaved cell stream. Accordingly, a line card will first examine this cell stream "on the fly" to ascertain important control information that will be used primarily to reassemble packets, and to format the reassembled packets for their destination interfaces. Additional processing of outbound packets is also conducted on the line card for PHY scheduling and bandwidth distribution purposes.

While the preferred mid-network processor 300 of the present invention has been described as delivering packets from a line card to an IPE card and then back to a line card (or to one or more additional IPE cards), the mid-network processor 300 can also be configured to route cells arriving over an ATM interface on one line card through the switch fabric and directly to another line card ATM interface, and can therefore function as an ATM switch.

Fig. 3 illustrates an exemplary line card 380 used in the preferred mid-network processor 300 of the present invention. As shown therein, the line card 380 preferably includes an ingress side (i.e., the left half of Fig. 3) and an egress side (i.e., the right half of Fig. 3). When packets are provided to the ingress side of the line card from the line card's associated PHY card, the packets are first provided to a packet inspector chip ("PIC") 400 which converts the packets

(which may already be represented by individual cells such as ATM cells) into fixed length cells. In this preferred embodiment, the fixed length cells are 64 byte cells that are 8 bytes wide and 8 bytes long. Thus, a "cell time," in the context of cells propagating within the preferred mid-network processor 300, corresponds to 8 clock cycles, as appreciated by those skilled in the art. The PIC 400 then examines the stream of fixed length cells "on the fly" to identify the "classification" (that is, the protocol encapsulation sequence), capture matrix, and other control information for each packet (as described more fully in copending Application No. 09/494,235 filed January 30, 2000 entitled "Device and Method for Packet Inspection," the disclosure of which is incorporated herein by reference). More specifically, the preferred PIC 400 generates a control cell for each examined cell of a packet, and each control cell represents the control information that has been determined thus far for the corresponding packet. Thus, the PIC 400 outputs both the stream of fixed length cells that was produced before this stream was examined "on the fly" therein, as well as corresponding control cells. As shown in Fig. 3, these control and data cells are then provided by the PIC 400 to four preferably identical buffer access controllers ("BACs") 402-408. Each BAC stores a different quarter (i.e., 25%) of the data cells received from the PIC 400 in its corresponding cell buffer ("CB").

Each control cell output by the PIC 400 also includes a protocol processing unit ("PPU") identifier which identifies a PPU associated with a particular BAC for processing that control cell. Note that each PPU, in this preferred embodiment, preferably comprises two

general purpose central processing units ("CPUs"), as shown in Fig. 3. Alternatively, a PPU could comprise one or more network processors, digital signal processors, or any programmable processors. The BACs 402-408 each examine the PPU identifiers contained in the control cells delivered thereto over a bus by the PIC 400. When a BAC determines that the PPU identifier in a particular control cell is identifying the PPU associated with that BAC, the BAC will forward the control cell to its associated PPU for processing, as described more fully below. Thus, while every BAC 402-408 in this preferred embodiment stores a quarter of every data cell in their associated cell buffers, each control cell output by the PIC 400 is acted on by only one BAC and its associated PPU. As a result of being so processed, the size of the control cell is much smaller than the typical size of a packet. This can significantly increase the utilization of the processor by reducing the I/O bandwidth which is the typical limiting factor in processor use. In this preferred embodiment, all control cells corresponding to a specific packet (and, more generally, to a specific user) are processed by the same BAC PPU on the line card 380.

Note that the PPU assigned by the PIC 400 for any given packet is performed according to configuration and control information received by the PIC 400 from a master PPU ("MPPU") 410, and can be changed by the MPPU 410 over time as necessary for PPU load balancing on the line card 380.

The PIC 400 also keeps track of the available memory addresses in the cell buffers associated with the BACs using a free buffer ("FB") list 412, and also keeps track of where each data cell is stored in the cell buffers

with respect to other cells of the same packet using a link list 414.

When a control cell is processed within a particular BAC PPU, the PPU produces a new control cell to be provided to a packet manager ("PM") 420 which is in communication with the PIC 400 and the BACs 402-408. Included in this control cell provided to the PM 420 is a dequeue pointer which designates the location of the first cell of a packet that is to be dequeued and sent to the PM 420 along with the second and subsequent cells of that packet (if applicable). The packet manager 420 then forwards this dequeue pointer back to the PIC 400, which, in turn, provides instructions to the BACs 402-408 to dequeue each quarter cell of the designed packet in sequence using the information previously stored by the PIC 400 in the link list 414. Thus, the designed packet is reassembled as it is dequeued and delivered to the packet manager 420.

At this point in the processing, the packet manager 420 stores the cells of the reassembled packet in its own cell buffer 422 (using a free buffer list 424 and link list 426). The packet manager 420 processes the control information it received for that packet from one of the BAC PPUs and then formats the packet according to this control information by modifying or augmenting the packet header as the cells of the packet are dequeued from the cell buffer 422. This process and additional details of the preferred packet manager 420 are described more fully in copending Application No. 09/494,236 filed January 30, 2000 entitled "Device and Method for Packet Formatting," the disclosure of which is incorporated herein by reference. The packet manager 420 also appends a header to each of the 64 byte cells that constitute the

reassembled and formatted packet, and these headers will be used by the switch fabric for routing the cells therethrough. The packet manager 420 then forwards the cells of the packet in sequence to a UDASL 430, which is provided for managing cell traffic into and out of the switch fabric for the line card 380. Typically, the UDASL 430 then forwards the packet cells into the switch fabric for delivery to an IPE card that will perform mid-network processing functions for the packet in question. This IPE card is preferably designated by the BAC PPU that prepared and forwarded control information to the packet manager 420.

Also shown in Fig. 3 is a 9-port Ethernet switch 450 which provides for interprocessor communications between the eight PPUs on the line card 380 (i.e., 4 PPUs on the ingress side and 4 PPUs on the egress side) and the MPPU 410 for purposes of load balancing, hardware monitoring and bandwidth distribution, and for sharing user and configuration information. The bandwidth distribution process and the preferred hardware are described more fully in copending Application No. 09/515,028 filed February 29, 2000 entitled "Method and Device for Distributing Bandwidth," the disclosure of which is incorporated herein by reference.

Fig. 4 illustrates an exemplary IPE card 500 used in the preferred mid-network processor 300 of the present invention. The hardware layout of the IPE card 500 is similar to the hardware layout on the ingress side (and the egress side) of the line card 380 shown in Fig. 3. That is, the IPE card 500 is also provided with a UDASL 501 that delivers a typically interleaved cell stream received from the switch fabric to a PIC 502. The PIC 502 is in communication with four BACs 504-510 that are

in communication with a PM 512. Thus, the primary difference between the preferred IPE card 500 and either side of the preferred line card 380 is the processing that is performed therein, even though this processing is performed with similar hardware. It should thus be apparent that the present invention provides, amongst other things, an inventive hardware module that can be programmed to perform requisite processing either on the ingress side or the egress side of a line card, or on an IPE card. This contributes to the configurability and scalability of the preferred mid-network processor 300, which can be reconfigured as necessary (both through programming and/or by adding additional lines cards and/or IPE cards) to accommodate additional users and/or to provide additional processing power.

Much like the PIC 400 resident on the ingress side of the preferred line card 380, the PIC 502 provided on the preferred IPE card 500 is used to inspect the stream of fixed length cells provided thereto by the switch fabric "on the fly" to ascertain control information for each packet to be processed on the IPE card. In most cases, this control information was added to the packet by the PM 420 on the ingress side of the line card that forwarded the packet to this particular IPE card. The PIC 502 outputs the stream of data cells to the four BACs 504-510, each of which is configured to store a different quarter of each data cell in its corresponding cell buffer (note that each BAC on the preferred IPE card 500 has two PPUs associated therewith, whereas only one PPU is associated with each BAC on the preferred line card 380). The PIC 502 also outputs control cells to the BACs 504-510, where each control cell contains a PPU identifier that designates one of the two PPUs associated

with a particular BAC for processing that control cell on the IPE card to perform mid-network processing functions for the corresponding packet. In this preferred embodiment, all control cells corresponding to a specific packet (and, more generally, to a specific user) are processed by the same BAC PPU on the IPE card 500.

For any given packet, the PPU that processed control information for that packet on the ingress side of the line card is also responsible for determining to which IPE card and, more specifically, to which PPU on a particular IPE card, the packet should be sent for further processing.

After a BAC PPU on the IPE card processes the control information for a particular packet, the PPU sends a control cell back to the PM 512, which then cooperates with the PIC 502 to dequeue the quarter cells of that packet in sequence from the cell buffers associated with the BACs 504-510. Upon receiving the constituent cells of a reassembled packet and storing these cells in its own cell buffer 514 (using a link list 516 and a free buffer list 518), the PM 512 processes the control cell received from the BAC PPU to format the reassembled packet according to its destination interface before forwarding the reassembled formatted packet back into the switch fabric for delivery to its destination line card (or another IPE card, in the case where additional processing of the packet is required).

Also shown in Fig. 4 is a 9-port Ethernet switch 550 which, like the Ethernet switch provided on the preferred line card 380, provides for interprocessor communications between the eight PPUs and an MPPU 530 on the IPE card 500 for purposes of load balancing, hardware monitoring

and bandwidth distribution, and for sharing user and configuration information.

Referring again to Fig. 3, it can be seen that the egress side of the exemplary line card 380 is also provided with a PIC 600, four BACs 602-608, and a PM 610. Upon receiving a possibly interleaved stream of fixed length cells from the switch fabric via the UDASL 430, the PIC 600 examines this cell stream "on the fly" to ascertain control information (including control information that may have been added to the packet header by the PM 512 on an exemplary IPE card 500). The PIC 600 then forwards the data cells to the BACs 602-608 for storage in their corresponding cell buffers, and forwards corresponding control cells for each packet to one of the BAC PPUs (typically assigned by an IPE card BAC PPU that previously processed control information for the same packet) for further processing. The assigned BAC PPU then performs additional packet processing, primarily for traffic shaping, PHY card scheduling and bandwidth distribution on that PHY card. This process and the preferred hardware are described more fully in copending Application No. 09/511,059 filed February 23, 2000 entitled "Method and Device for Data Traffic Shaping," the disclosure of which is incorporated herein by reference. Upon processing the control information received from the PIC 600, this BAC PPU produces and forwards a control cell to the packet manager 610, which, in turn, dequeues the quarter cells of the corresponding packet in sequence from the cell buffers associated with the BACs 602-608 in cooperation with the PIC 600. The PM 610 then stores the constituent cells of the reassembled packet in its own cell buffer 612 (using a link list 614 and a free buffer list 616), and formats the packet for

its intended destination before forwarding the reassembled formatted packet to the PHY card associated with this line card for outputting the packet from the mid-network processor 300.

A description of one preferred implementation of the broadband mid-network server described above will now be provided, wherein the following terms have the following meanings:

CardId: An 8 bit number that uniquely identifies an IPE or Line Card in the system.

FlowId: A 10 bit number whose lower (least significant) 8 bits contain a CardId, and whose upper (most significant) 2 bits identify the priority (class) of the traffic sent through the switch fabric to this card using this FlowId. (In the switch fabric, this field is 12 bits, but our implementation only uses the least significant 10 bits.)

User: A datalink (layer 2) interface. Examples include ATM virtual circuits, PPP sessions (over SONET, Ethernet, or ATM), and MPLS label switched paths.

UserId: A 32-bit value that can be used as a system-wide pointer to user configuration and state information. Since multiple cards (one or more IPEs and one Line Card) can store information about a user, it is possible to have multiple UserIds that refer to a single user. The upper (most significant) 8 bits of the value represent the CardId of the card which contains the user information being identified. The next 4 bits represent the PPUID of the PPU on the card where the information is stored, and the lower (least significant) 20 bits represent the CID assigned by that card to the user. The CID is used as an index into the PPU's table of user information.

LCUserId: A UserId in which the CardId identifies a Line Card.

Primary UserId: A UserId in which the CardId and PPUID identify the PPU on an IPE with has the primary responsibility for managing a user.

Secondary UserId: A UserId in which the CardId and PPUID identify an IPE PPU other than the one identified by the Primary UserId

Small User: A user whose ingress packet stream is processed entirely by a single IPE PPU. Small users do not have Secondary UserIds.

Large User: A user whose configured bandwidth is too high for his ingress packet stream to be processed by a single IPE PPU. All large users have one or more Secondary UserIds.

Logical Link: A group of users of the same type (i.e.: a group of ATM Virtual Circuits). If the Logical Link is a group of PPPoE sessions over ATM, the Logical Link must be an ATM Virtual Circuit.

CSIX Header: The header of a CSIX (i.e., Common Switch Interface) cell. The CSIX Header is separate from the 64 byte cell payload.

Cell Header: The first two bytes of the 64 byte payload of a CSIX cell.

PIE Header: The 6 bytes immediately following the Cell Header of the first cell of a packet.

Overview:

In this particular implementation, the server system preferably comprises one or more rack mountable system units (i.e., shelves). The system also contains at least one line card, exactly as many PHY cards as line cards, and at least as many IPE cards as line cards. Also, each shelf of the system contains preferably three switch fabric cards and two flash disk cards. Each line card is uniquely associated with a particular PHY card. However, there is no particular association between line cards and IPE cards.

Each IPE card can be thought of as an independent router, with one or more IP addresses associated with it. Each Layer 2 (datalink) interface (referred to as a "user") provided by a line card is associated with exactly one IPE card (more specifically, exactly one PPU on one IPE card). Different users from the same line card can be associated with different PPUs on different IPE cards, and a particular PPU can have users from multiple line cards.

Since it is possible for multiple Layer 2 protocols to be encapsulated within each other (for example, PPP/Ethernet/ATM), there is an exception to the "one user, one PPU" rule. In this case, the inner-most levels of encapsulation, each of which being layer 2 interfaces (users) in their own right, can be associated with different PPUs within an IPE card, or even PPUs on different IPE cards, thus causing traffic from the outer levels of encapsulation to be split among multiple PPUs or IPE cards. It is also possible for outer layers to be encapsulated layer 3 traffic as well as layer 2 traffic (for example, an Ethernet/ATM virtual circuit can carry IP as well as PPPoE packets). In this case, all the layer 3 traffic will be associated with a single PPU (a user), but the encapsulated layer 2 datalinks (users) can each be associated with a different IPE card.

The set of all users on the system is preferably distributed as evenly as possible across all the IPE cards in the system. Within an IPE, the MPPU stores the per-user information for the users assigned to that IPE and distributes those users across its PPUs. Each PPU stores a copy of the per-user information assigned to it. Thus each user is associated with one and only one IPE card and one and only one PPU on that IPE. This PPU's copy of the user's configuration and state information can be uniquely identified on a system-wide basis by the Primary UserId.

The architecture of this preferred implementation is based on line cards, PHY cards, a switching fabric, internet processing engines (IPE) and flash memory modules, as was

described generally above. The line cards terminate the link protocol and distribute the received packets based on user, tunnel or logical link information to a particular IPE through the switching fabric. The procedure of forwarding a packet to a particular IPE and PPU will be denoted as "routed distribution." A midplane is also used to connect the different cards. The preferred line card and the preferred IPE card were described above with reference to Figs. 3 and 4.

The system is comprised of a set of hardware components, as described, which can be used to configure a system for a wide variety of applications as well as throughput requirements cost effectively. The preferred switch fabric and scheduler support cell switching at OC-192 speeds, and the switch fabric is both fully redundant and highly scalable. The preferred IPE cards have the following attributes: high performance protocol processing engine; manages users, tunnels and secure segment groups; supports policing and traffic shaping; implements highly sophisticated QoS with additional support for differentiated services; supports distributed bandwidth management processing; and supports distributed logical link management, able to do NAT, packet filtering and firewalls.

The preferred line cards have the following attributes: packet lookup processing; protocol identification; scheduling; supports distributed bandwidth management processing; multi-I/F support (ATM, GE, POS); and AAL-5 Processing (CRC check and generation).

The preferred PHY cards have the following attributes: line termination for rates up to OC 192c; ATM - Layer Processing; ATM - SONET Mapping; POS - SONET Mapping (including FEC checksum computation); GE - MAC and PHY Processing; and support the following line cards: ATM: 4x OC-48, 8x OC-12, 16x OC-3; POS: 1xOC192, 4x OC-48, 16x OC-12; and GE: 8/10x GE. Additionally, the overall system preferably has the following attributes: high availability; 1+1 switch fabric and scheduler redundancy; 1+1 control system unit

redundancy; all field replaceable units are hot-swappable; N+1 AC power supply redundancy; and N+1 fan redundancy.

One purpose of routed distribution is to forward a packet to a particular PPU within an IPE. The key benefits of this approach are: incremental provisioning of compute power per packet; allows load distribution based on the packet computation needs for a particular user or tunnel; user and tunnel configuration information can be maintained by one single processor thus minimizing the inter-process communication needs; and allowing the portability of single processor application S/W onto the system.

Fig. 5 illustrates the distribution of packets to a particular IPE. A packet is received from a line card. The line card examines the packet and forwards the packet based on the IP source or destination address, the user session ID, or the tunnel ID. The IPE receives the packets and hands it over to the PPU specified by the line card.

The line cards and the IPE host the flexible protocol-processing platform. This platform is comprised of a data path processing engine and the already mentioned protocol-processing unit. The separation of data path processing from protocol processing leads to the separation of memory and compute intensive applications from the flexible protocol processing requirements. A clearly defined interface in the form of dual-port memory modules and data structures containing protocol specific information allows the deployment of general-purpose CPU modules for supporting the ever changing requirements of packet forwarding based on multi-layer protocol layers.

The protocol-processing platform can be configured for multiple purposes and environments. That is, it supports a variable number of general purpose CPUs which are used in the context of this architecture as Protocol Processing Units (PPU). One of these CPUs is denoted as the Master Protocol Processing Unit (MPPU).

The data path processing unit extracts, in the packet inspector, all necessary information from the received packets or cells and passes this information on to a selected PPU via one of the buffer access controller devices. The cells themselves are stored in the cell buffer and linked together as linked lists of cells, which form a packet. Once a PPU has selected a packet for transmission, it passes the pointer to the packet and the necessary formatting and routing information to the data path processing unit. This enables the formatting and the segmenting of the packet. The packet is then forwarded either as a whole or segmented based on the configured interface.

Each PPU is associated with one dual-ported memory, where one port is controlled by the data-path processing unit and the other by the corresponding PPU. Each dual-ported memory contains two ring buffers, where one ring buffer is used to forward protocol specific information from the data path to the PPU and the other is used for the other direction. The ring buffer for passing on protocol specific information to the PPU is called the receive buffer. The other buffer is called the send buffer. Two pointers are maintained for each ring buffer. The write pointer for the receive buffer is maintained by the data path processing unit while the read pointer is set by the PPU. The send buffer's write pointer is controlled by the PPU and the read buffer by the data path processing unit.

The PHY Card:

The PHY card terminates the incoming transmission line. It also performs clock recovery and clock synthesis. Optical signals are converted into a parallel electrical signal which is then an input to a physical framer device which maps the incoming bit stream into the transmitted physical frame. Finally the physical layer of the corresponding link protocol processes the physical frames. In addition, link layer protocol processing is performed in order to provide a common packet interface to the line card. On the transmission side,

the packets or cells are mapped into physical frames. These frames are then encoded into the corresponding physical layer format and sent over the optical fiber to the receiving peer. The physical layer format is preferably either SONET or Gigabit Ethernet. The link layer format is preferably GE, ATM or PPP for POS.

The Line Card:

The line card performs packet forwarding for the egress and ingress path. Full duplex 10 Gbit/s throughput is provided. The line card interfaces to the PHY cards and the switch fabric card. The Line Card is preferably configured for either POS-PHY or UTOPIA III interface to the PHY card. The Line Card preferably hosts two Protocol Internet Engine (PIE) chip sets. On the ingress side, one PIE chip set supports four protocol-processing units (PPU) and one MPPU. The Four PPUs perform routed distribution to the various IPEs in the system. They also provide traffic shaping and scheduling of flows to the switching fabric. The remaining MPPU is used for overall control and supports the distributed bandwidth allocation protocol of the switching fabric.

The Packet Inspector (PI) first examines incoming cells or packets and the protocol information is extracted based on matched patterns in the data flow. This information is then made available to the PPU which is responsible for processing the incoming packet. Cells or packets from a PHY card are processed by a particular PPU based on a chosen configuration. This configuration depends upon the configuration of the PHY card itself and upon the protocol supported by the PHY card.

The other PIE chip set, processing the egress flow, is preferably responsible for cell assembly from the switch fabric and packet scheduling for multiple physical ports. Additional support for AAL5 processing is provided for ATM flows. The MPPU from the ingress path is shared for configuration, maintenance and cell extraction of the egress flow.

The communication channel provides signaling and connection setup control for the ATM PHY card. The PHY card informs the Line Card about the physical layer status and reports alarm and error conditions.

The ingress packet processing preferably involves: Packet Assembly for ATM traffic (AAL5 processing); Protocol Identification (Packet Data Inspection); Routed Distribution; Scheduling of traffic flows through switching fabric; Buffer management for ingress cell buffers; and cell scheduling for the switch fabric.

The egress packet processing preferably involves: Traffic Shaping; Packet Assembly for switch fabric flow; MPHY Buffering; Cell Scheduling for ATM with multiple physical interfaces with AAL5 processing (CPCS, SAR); and Packet Scheduling for POS with multiple physical interfaces.

The Internet Processing Engine (IPE) provides the functionality for protocol processing, user management, tunnel management and secure segmentation. It receives the packets from the switching, enforces the service level agreements (SLA's), performs packet classification, filtering and forwarding, and finally schedules the packet for transmission to the requested interface.

The PI is part of the Packet Internet Engine (PIE) chip set, which consists of the Packet Inspector, the Buffer Access Controller, and the Packet Manager. Together with the sixteen PPUs and the MPPU, the PIE chip set provides a powerful Protocol Processing unit. The PIE chip extracts informative protocol information and forwards it to the PPUs and the MPPU based on the routed distribution decision made in the Line Cards. The chosen PPU processes this information and performs all necessary packet processing. This includes, besides forwarding and filtering, policing, and packet formatting. The MPPU controls the IPE and is negotiating with the units in the system the bandwidth allocation of the switch fabric. It also provides bandwidth management for the configured logical links. The MPPU manages its connections by assigning users and

tunnels to individual PPUs for forwarding processing from the Line Card to a particular IPE. Once a connection between the MPPU and Line Card is set up, all packets belonging to such a connection are forwarded from the Line-Card to the chosen PPU.

A PPU is chosen based on the already assigned connections, their bandwidth and the bandwidth and QoS required for the new connection. Connectionless traffic (Internet to Internet) is mapped onto an internal connection. If more bandwidth is needed than one PPU can manage, the packets will be distributed over multiple PPUs.

The functionality of the IPEs include: User Management; Tunnel Management; Logical Link Management; Support for Secure Segmentation; Policing; QoS Control with Diff Service Support; Buffer Management; IPv4, IPv6 Forwarding; Packet Classification; Packet Filtering with support for user Filters; Celox Management Database Support; Packet Formatting, and NAT.

The Protocol Internet Engine Chip Set (PIE):

The Protocol Internet Engine (PIE) provides the data path processing capabilities for the server system at OC-192c rates. The PIE chip set comprises three chips. These chips result in a very high performance packet processing system together with an interface controller and multiple general purpose CPUs.

Each cell is preferably transferred into the buffer through four buffer access controllers ("BACs") in order to increase the bandwidth to the PPUs and to increase the bandwidth to the external cell buffers. Different portions of the same cell are written to the cell buffers attached to the different BACs. However, the captured portion of the data is sent to just one of the PPUs.

The preferred BAC unit is shown in Fig. 8. The RSU receives incoming data, reformats the data to an internal format, performs a parity check for incoming data, and also performs synchronization control. The preferred format of a cell received by the BAC from the packet inspector is shown in

Fig. 9. Referring again to Fig. 8, the Cell Filter unit extracts control information from the cell and sends the cell data to the BAU along with the indication of which portion of the cell has to be stored in this cell buffer. The CFU also sends the cell data stream to the PTU which translates the PPUID to the appropriate PPU and thence to the CCU where, based on the PPUID and the capture matrix, the control cell is extracted from the data cell CCU and stored in the CBU. The CMU then transmits the control cell to the appropriate PPUs through a dual port RAM interface.

When the packet has to be dequeued, the control cell corresponding to the packet is sent by the PPU which processed that user to the PM along with the dequeue pointer. This is received by the BEU of the PM, as shown in Figure 8.

The control cell data stream (shown as the narrow arrow in Fig. 8) then goes to the ICU where it is stored while the DSU does deficit round robin scheduling of the data packets corresponding to the control packets in order to distribute bandwidth equitably to the BACs for sending out packets. In addition, the dequeue pointer corresponding to the packet to be dequeued is sent to the PIU from where it is transmitted to the PI where it is received at the PIU and passed on to the BMU. In the BMU, the dequeue pointers are stored in a FIFO while the previous packets are being dequeued. The dequeue pointer information is passed onto the BACs and the BAU in the BACs dequeues the packet and passes it through the PMU to the packet manager. A packet is dequeued by dequeuing all the cells comprising the packet which are held in the form of a linked list. Data packets from the data packet stream (shown as the thick arrow in Fig. 8) undergo AAL5 processing (should they need it) in the APU, and are stored in the IDU buffer. The FAU reformats packets into 64 bit slices and controls dequeuing from both the IDU and the DSU's DPRAM in accordance with the PFU. In order to ensure matching of the control packet with the data packet, a sequence number is used at the beginning of both the data and the control cells. Both the

control and data streams enter the PFU where they are formatted and sent to the TIU to be sent to the phy cards or the switch fabric.

The PIE chip set can be configured for multiple purposes and environments. That is, it supports a variable number of general purpose CPUs which are used in the context with the PIE chip set as Protocol Processing Units (PPU). One of these CPUs is reserved for maintenance and control purposes and is denoted as MPPU.

The PIE chip set implements all necessary functions in order to hide all data path processing from the actual protocol processing functionality. The PIE chip set extracts all necessary information from the received packets or cells and passes this information on to a selected PPU. The cells are then stored in the cell buffer and linked together as linked lists of cells, which form a packet. Once the PPU has selected a packet for transmission, it passes the pointer to the packet and the necessary formatting and routing information to the PIE chip set. This allows formatting and segmenting of the packet. The packet is then forwarded to the MPHY scheduler as a whole or segmented based on the configured interface.

Each PIE chip set is differently configured. The PIE chip set on the IPE supports as many as 8 PPUs and 1 MPPU. 4 PPUs and 1 MPPU will support the PIE chip set on the ingress side of the Line Card, and an equal number on the egress side of the Line Card.

The characteristics of the preferred PIE are as follows: Three Chip Chip-Set; Full Data-path processing in hardware; Support for distributed protocol processing by general purpose CPU modules; Highly scalable compute power per packet (up to 64 PPUs can be supported); Flexible interface support with MPHY scheduling; AAL-5 Processing; SAR Sublayer: Assembly and Segmentation for up to 256K connections; CPCS Sublayer: CRC 32 generation and check, padding control, and length field control; Internal Packet Processing; Checksum computation and

check; Length field control; Padding control; Micro-programmable Packet Inspection Engine; Supports any layer packet inspection; Supports byte matched pattern processing; Supports bit matched pattern processing; Results are made available to protocol processing units; Supports extraction of any portion of packet for protocol processing; IPv4/IPv6 Header Checksum; Congestion Avoidance Support; EPD; PPD; Internal Back-pressure control; Linked List Control; Supports up to 8 million 64 byte cells (initially a million); Links cells together to form a packet; Garbage Collection; Assembly aging control; Buffer Access; Parity generation and check for signal integrity; Cyclic access for data rates up to 12Gbit/s; PPU Access; Dual-Port access control for up to 8 dual-port RAMs each with 512/256 KByte memory; Support for dual-port RAM data synchronization; Dual-ring buffer control for each dual-port RAM for data exchange; Threshold-based access control for writes to ring buffer; Support for up to 24 Gb/s throughput (bi-directional); Back-pressuring in case of buffer overflow; Cyclic Packet Scheduling; Packet Scheduling for cyclic access control with support for data rates up to 12Gb/s; Micro-programmable packet formatter; Supports insertion, removal and overwriting for any byte in a packet at OC-192 speeds; Supports IPv4/IPv6 Header Checksum generation; Support UDP/TCP checksum generation; Cell Scheduling, Buffering and Linked List Management; Supports cell buffering for up 512K cells; and supports scheduling for up to 1024 queues.

Together with the PPUs, the preferred PIE supports:

Packet Classification:

Based on Layer 3,4,... Information (any layer); Packet Filtering; User programmable filters; Group filters; Firewall processing; Packet Forwarding; IPv4 Lookup Processing; IPv6 Lookup Processing; Tunnel Forwarding; Buffer Management; Dynamic Thresholding on a per user and assigned rate basis; Support for up to 8 million Cell Buffer (initially a million); Congestion avoidance with Early Packet Discard (EPD), Partial Packet Discard (PPD), Selective Packet Discard; Policing; Per

User and Logical Link; Enforcing traffic contracts based on SLA; Traffic Shaping; Per User and Logical Link; Support for traffic contracts based on SLAs; Support for Real-time traffic (low delay traffic); QoS Control; Supported for differentiated services; Multiple priorities per user; Flow based queuing (not initially supported); Bandwidth Management; Distributed processing for allocation of bandwidth on switch-fabric links including MPHY links; Distributed processing for allocation of bandwidth for logical link management; User Management for up to 512K users; Tunnel Management for up to 128K users; L2TP; IPSec; Multi Protocol Processing; and Support for any protocol.

Traffic Management:

Traffic Management for an Internet access system is complex due to the involvement of various system interfaces. A system might be connected to users, the Internet backbone, a Local Area Network with file and Web servers, and a Metropolitan Area Network (MAN) which gives access to local TV and media servers as shown in Fig. 11. Each link has different link properties with respect to available bandwidth and Dollar per Megabyte. This means that a user's share of bandwidth on a particular link has to be based on the property of this link. A user might get more bandwidth share on the MAN link than on the backbone link due to the fact that more bandwidth at a cheaper price is available on the MAN link than on the backbone link. The same is true for bandwidth wholesaling of the preferred system to multiple ISPs who would like to resell bandwidth to their customers. The enabling technology for this model is Secure Segmentation. This model has also led to the introduction of logical link groups. A logical link group can be assigned to a secure segment based on the bandwidth needs of the considered secure segment for a particular link as shown in Fig. 12. This means that not only user allocation has to be considered but also logical link bandwidth needs to be included. Therefore, bandwidth is distributed based on traffic class, user, and logical link

group. This supports the wholesaling model and takes into account over-subscription requirements in order to support QoS including differentiated services.

The preferred system represents a highly distributed system. In such a system, resources have to be allocated based on the requirements of the traffic of each component. That means in general that each component has to take part in a distributed computation method in order to allocate the resources. The traffic management requirements for bandwidth allocation within the preferred system will have to include bandwidth negotiation for the various flows through the switching fabric. One also has to consider the specific requirements to support the above-introduced concept of logical link groups. Since logical link groups are managed in a distributed manner, bandwidth information has to be exchanged between the entities managing one logical link as shown in Fig. 13. Buffer management and QoS Control is an integral part of the overall traffic management scheme implemented in the preferred server system. Due to the large buffer, the system has to maintain on various different places in the distributed system a sophisticated buffer management scheme which has to be implemented and supported by QoS control in order to support differentiated services and other traffic flow specific requirements

Policing - Traffic Shaping:

Policing and Traffic Shaping have closely related functionality. Policing ensures that the incoming stream does conform to the negotiated link parameters for a logical link group as well as the user of the incoming link. Traffic shaping enforces the link parameters for the outgoing traffic stream based on the outgoing user, the logical link group and the link itself. Fig. 14 is intended to illustrate the need for policing as well as traffic shaping. An incoming traffic stream is shaped (policed) in order to enforce the traffic contracts of a user for the considered link and logical link. Before the traffic is forwarded to another link, the traffic

contracts for this particular link have to be enforced. This traffic contract might be much different from the traffic contract of the incoming stream. Consider the case where a user requests information over the Internet backbone link. The bandwidth allocated on this link for this user might be 500 Kbit/s. The logical link bandwidth for the corresponding secure segment might be set to 10 Mbit/s. If the user's access link to the system uses an ATM connection with an assigned rate 1 Mbit/s and no policing is enforced, the user could use the full 1 Mbit/s. This is possible since the traffic shaped onto the user ATM link allows the user to transmit the higher rate. Therefore, it is necessary to police the incoming traffic and the other for shaping the traffic for a particular link.

Fig. 15 shows the schematic implementation of the policer and traffic shaper in an IPE within the preferred server system. A received cell is assigned to a particular user data structure assigned to the incoming link for the considered user. As discussed earlier, the policing information can be directly obtained from the user who is sending a packet based on the connection identifier, the corresponding session ID, or the IP source address. However, if the packet on the incoming connection cannot be directly associated with a user or logical link group, then the user and/or logical link group for whom it is destined classifies the packet. Based on the obtained user and logical link information the incoming traffic stream is policed by queuing up the packets and enforcing the negotiated traffic contract.

Once the packet conforms to the incoming link requirements, the packet is shaped based on the user parameters and logical parameters for the outgoing link. These parameters are obtained from the user connection itself if a session ID can be associated with it. If the packet comes from a user and is forwarded across the Internet to a remote terminal, then the shaping parameters are obtained from the sending user for the corresponding link and the associated

logical link group. For connectionless traffic, which cannot be directly associated with users, logical link group can be assigned based on the IP destination address and or source address. This allows managing traffic flows between networks.

Switch Fabric Bandwidth Management and Scheduling:

In order to meet the QoS requirements of individual traffic flows and to ensure that delay requirements of certain flows can be met, sophisticated scheduling must be conducted across the entire switch fabric. This scheduling takes into account the allocated user bandwidth, logical link share, buffer occupancy for output queues, available sub-port bandwidth, priority of class of traffic, and expected delay. All this is accomplished while maintaining high throughput across the switch fabric.

Attached hereto as Exhibit A are details of the manner in which the preferred server system is programmed so as to minimize inter-IPE card communications.

There are various changes and modifications which may be made to the invention, as apparent to those skilled in the art. However, such changes and modifications are suggested by the present disclosure, and the invention should therefore be limited only by the scope of the claims appended hereto, and their legal equivalents.

EXHIBIT A

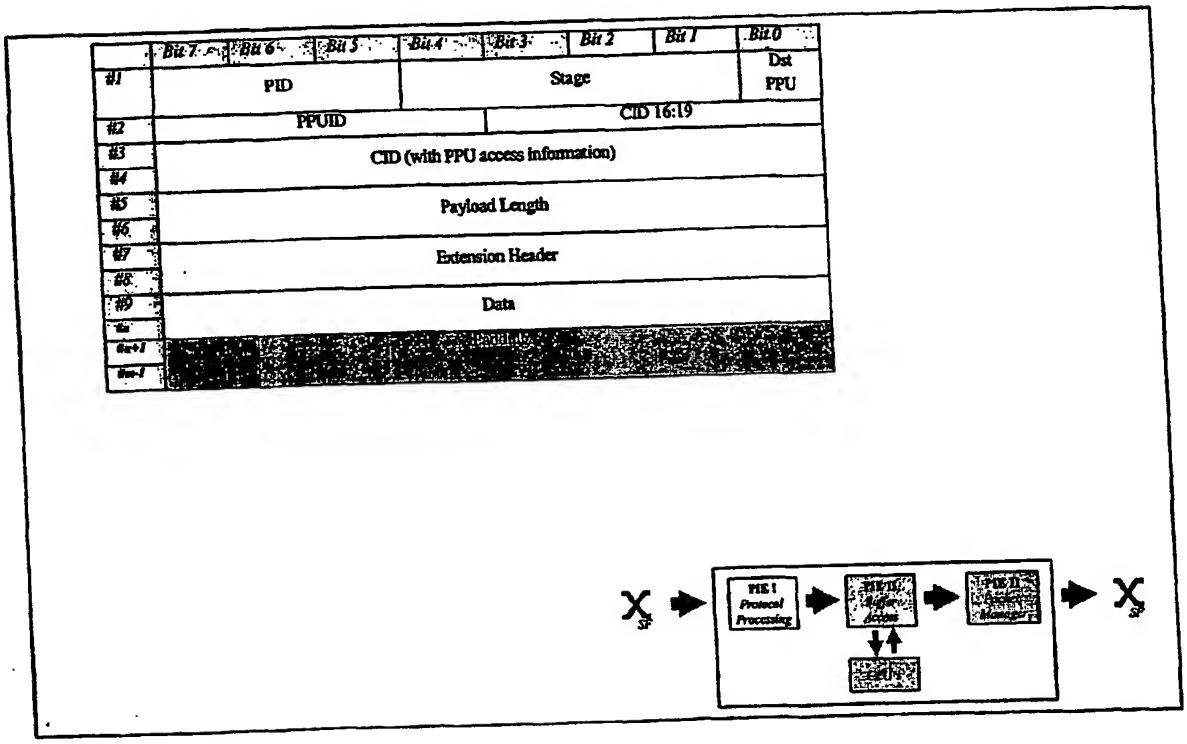
1.1.1 *Line Cards – Ingress Side*

Line Cards do not perform any traffic policing. Policing is performed, in distributed fashion, by the all IPE cards in the system. If during testing, it can be determined that the Line Cards have enough processor and I/O bandwidth to perform policing, this function might be moved to the Line Cards in a future version of the software. Also, Line Cards do not perform any routing table lookups.

All IP packets received, regardless of their encapsulation, must have their destination IP address captured and examined by a LC PPU. One operation that must be performed is determining if the destination IP address is one of the IP addresses of our system. This can be done using a simple hash table. A full CIDR routing search is not necessary, since we are only looking for an exact match. The result of the lookup (if successful) is the CardId of the IPE that the address belongs to. If a match is found and the CardId is equal to the CardId of the IPE that the packet is about to be forwarded to, the packet must be forwarded with the Destination PPU bit set. This is so that when the packet is received, the PI can select the packet to be captured in its entirety (as long as it is not part of a non-encrypted tunnel).

Additionally, if the packet is an IPsec packet that has been received from a *large user*, and the destination IP address is one of the addresses of the IPE to which the packet is about to be forwarded to, the UserId should be determined based on the IPsec Security Parameter Index (SPI) rather than on the hash of the source and destination IP addresses in the IP header of the packet. These operations will be discussed in greater detail in the sections that follow.

PIE Header



1.1.2 ATM Line Card

1.1.2.1 Small Virtual Circuits

The following information is sent to the IPE PPU along with the packet payload:

- In the *CSIX Header:*
  - Destination FlowId:
    Sent in the *CSIX Header* of every cell of the packet to identify where the switch fabric should send it as well as the priority (class) of the packet.
- In the *Cell Header:*
  - Source FlowId:
    Sent in the *Cell Header* to allow the IPE to reassemble the packet. This is simply the identification of the line card (in the least significant 8 bits) and the priority (class) in the most significant two bits. The priority MUST be the same as is specified in the Destination FlowId of this packet.
  - Discard Bit:
    Set by the LC PM to inform the IPE that an error (IP checksum, AAL5 CRC, internal parity) was detected in the packet.
  - EOP Bit:
    Set by the LC PM to indicate the last cell of the packet.
- In the *PIE Header:*
  - Initial PID:
    This 3 bit field tells the IPE the type of encapsulation this packet has. The choices are: IPC (for inter-processor communications), IP, PPP, Ethernet, ATM, or MPLS
  - Initial Stage:
    This 4 bit field can be used to give additional information to the IPE about the encapsulation of this packet. It specifies which stage in the IPE PI will be the first to inspect the packet.
  - Destination PPU (1 bit):
    This bit is set for IP packets whose destination address is equal to one of the IP addresses of the IPE card that packet is being sent to.
  - Destination PPUID:
    The PPU identifier of the IPE PPU that the packet is being sent to.
  - IPE CID:
    An index into the connection table of the IPE PPU that the packet is being sent to.

The PI uses the VPI/VCI and PHYID to calculate the LC CID, which is used by the PI as index into the hardware connection table. The PI reads (amongst other things) a LC PPUID which selects the LC PPU that the control information for the packet should be sent to.

The LC CID is also used by the LC PPU as an index into a software connection table. Typically (though not always), this connection table is used to determine the *UserId* (which consists of a Destination CardId, Destination PPUID, and IPE CID) that is sent to the IPE in the *PIE Header* of the packet. As packets are inspected by the LC, a determination of priority (one of four classes) is made based on the protocols found in the packet. Alternatively, the priority could be read from the connection table. This priority is used to determine the two most significant bits of the Destination FlowId when the packet is forwarded to an IPE.

The ATM cell headers and the AAL5 trailer and padding are removed (by the PM) before forwarding the packet.

The following is a list of all the different types of top level protocol encapsulation that can be received by the ATM line card, along with an explanation of the processing that must take place on the line card for each type of protocol:

1.1.2.1.1 IP (RFC-1483)

The IP packet is forwarded to the IPE. The IPE CID is determined by reading the software connection table.

1.1.2.1.2 Ethernet (RFC-1483)

Each ATM LC must have a standard globally unique Ethernet MAC address permanently assigned to it. Each Ethernet/ATM VC should be configurable as to whether or not it is in "promiscuous" mode – that is, whether or not it should discard unicast packets not sent to its MAC address.

All Ethernet packets are forwarded to the IPE with their MAC headers intact, except for PPPoE session packets (ethertype = 0x8864). For non-PPPoE session packets, the IPE CID is determined by reading the software connection table, and the Initial PID is set to indicate an Ethernet packet.

For PPPoE session packets, the PPPoE header is removed, and the PPPoE Session ID (from the PPPoE header) is used to index into a PPPoE session table, from which the IPE CID can be retrieved. In this case the Initial PID is set to indicate a PPP packet. Additionally, if the PPP/PPPoE protocol type is IP, the PPP header is also removed before the packet is forwarded to the IPE, and the Initial PID is set to indicate an IP packet.

1.1.2.1.3 PPP

If the PPP protocol type is IP, the PPP header is removed, and the Initial PIDS is set to indicate IP, otherwise, the PPP header is kept, and the Initial PIDS is set to indicate PPP. The IPE CID is determined by reading the software connection table.

1.1.2.1.4 MPLS

The top of stack shim label (in the AAL5 PDU) is replaced with the VPI/VCI of the virtual circuit. The VPI/VCI can be deduced from the LC CID. The IPE CID is determined by reading the software connection table.

1.1.2.2 Large Virtual Circuits

With the exception of the following changes, large virtual circuits are handled in the same way as small virtual circuits.

The PI DFU control registers can be programmed (by the MPPU) with the LC CID's of up to 4 large virtual circuits. For these circuits, if the packet contains an IP header, the PI DFU will replace the LC PPUID read from the hardware connection table with a LC PPUID read from a hash table which is indexed by a hash of the source and destination IP addresses of the packet (calculated by the PI DFU).

Any of the entries (circuits) in the software VC connection table can be marked for distribution across multiple IPE PPUs. These are known as *large users*, and need not be the same virtual circuits that are distributed by the DFU as explained above. For these circuits, if the packet contains an IP header, a new hash is calculated over the source and destination IP addresses of the packet and used to select one of several *UserIds* (Destination CardId, Destination PPUID, and IPE CID) that are sent to the IPE in the PIE Header of the packet. The most significant bits of the Destination FlowId are still used to select the priority (class) of the packet. However, in the case of an IPsec packet addressed to the IPE card that the packet will be forwarded to, the *UserId* is selected using a different means, as described in the IPsec protocol processing section below.

1.1.3 POS Line Card

The following information is sent to the IPE PPU along with the packet payload:

- In the *CSIX Header:*
  - Destination FlowId:

Sent in the *CSIX Header* of every cell of the packet to identify where the switch fabric should send it as well as the priority (class) of the packet.

- In the *Cell Header:*
  - Source FlowId:
    Sent in the *Cell Header* to allow the IPE to reassemble the packet. This is simply the identification of the line card (in the least significant 8 bits) and the priority (class) in the most significant two bits. The priority **MUST** be the same as is specified in the Destination FlowId of this packet.
  - Discard Bit:
    Set by the LC PM to inform the IPE that an error (IP checksum, internal parity) was detected in the packet.
  - EOP Bit:
    Set by the LC PM to indicate the last cell of the packet.
- In the *PIE Header:*
  - Initial PID:
    This 3 bit field tells the IPE the type of encapsulation this packet has. The choices are: IPC, IP, PPP, or MPLS
  - Initial Stage:
    This 4 bit field can be used to give additional information to the IPE about the encapsulation of this packet. It specifies which stage in the IPE PI will be the first to inspect the packet.
  - Destination PPU (1 bit):
    This bit is set for IP packets whose destination address is equal to one of the IP addresses of the IPE card that packet is being sent to.
  - Destination PPUID:
    The PPU identifier of the IPE PPU that the packet is being sent to.
  - IPE CID:
    An index into the connection table of the IPE PPU that the packet is being sent to.

Unless MPLS/PPP/SONET is being used, each PPP/SONET PHY comprises a single *user*. When MPLS is in use, however, each MPLS Label Switched Path (LSP) represents an additional *user*.

For POS Line Cards, the LC CID is simply the PHYID. The PI DFU control registers can be programmed (by the MPPU) with the LC CID's of up to 4 PHYs. For these PHYs, if the packet contains an IP header, the PI DFU will replace the LC PPUID read from the hardware connection table with a LC PPUID read from a hash table which is indexed by a hash of the source and destination IP addresses of the packet (calculated by the PI DFU). This capability of the PI DFU must be used for OC-192c and OC-48c PHYs in order to distribute the load over multiple LC PPUs. For OC-12c and smaller PHYs, the PI DFU need not be used. Instead, the PI uses the LC CID as index into the hardware connection table. The PI reads (amongst other things) a LC PPUID which selects the LC PPU that the control information for the packet should be sent to.

As packets are inspected by the LC, a determination of priority (one of four classes) is made. This priority is used to determine the two most significant bits of the Destination FlowId when the packet is forwarded to an IPE.

The following is a list of all the different types of top level protocol encapsulation that can be received by the POS line card, along with an explanation of the processing that must take place on the line card for each type of protocol:

1.1.3.1 PPP Control Protocol (LCP, PAP, CHAP, IPCP, MPLSCP, etc...)

This category includes not only PPP Control Protocols, but also any Network Protocol other than IP or MPLS.

The LC PPU uses the LC CID (which is really just the PHYID) as an index into a software PHY table. This table provides the *Primary UserId*, which determines where the packet is sent as well as the IPE CID that is sent in the *PIE Header* of the packet. For large and small PPP/SONET *users*, all non-IP and non-MPLS packets are sent to the IPE PPU identified by the *Primary UserId*. No distribution is performed for these packets. Also, the Initial PID is set to indicate a PPP packet.

1.1.3.2 IP

The LC PPU uses the LC CID (which is really just the PHYID) to index into and read from the software PHY table. From this the LC determines whether this is a *small user* or a *large user*. For *small users*, the *Primary UserId* is also read from the PHY table. This determines where the packet is sent as well as the IPE CID that is sent in the *PIE Header* of the packet.

For *large users*, however, a hash is calculated over the source and destination IP addresses of the packet and used to select either the *Primary UserID* or one of several *Secondary UserIDs*. The selected *UserID* determines where the packet is sent as well as the IPE CID that is sent in the *PIE Header* of the packet.

The PPP header is removed before the packet is forwarded to the IPE, and the Initial PID is set to indicate an IP packet.

1.1.3.3 MPLS

As is the case with IP/PPP/SONET, the LC CID only identifies the PHYID. Therefore, when the LC PI identifies an MPLS packet, the top of stack label must be captured in order to identify the *user*. For each POS PHY, the LC PPU must maintain a table of MPLS LSPs. The LC CID selects which table, and the top of stack label is used to index into the table. For *small users*, the *Primary UserID* that corresponds to the LSP can then be read the table. For *large users*, however, a similar process to the one described above for IP is used. A hash is calculated over the source and destination IP addresses of the packet and used to select either the *Primary UserID* or one of several *Secondary UserIDs*. The selected *UserID* determines where the packet is sent as well as the IPE CID that is sent in the *PIE Header* of the packet.

The PPP header is removed before the packet is forwarded to the IPE, and the Initial PID is set to indicate an MPLS packet.

1.1.4 Ethernet Line Card

The following information is sent to the IPE PPU along with the packet payload:

- In the *CSIX Header:*
  - Destination FlowId:
    Sent in the *CSIX Header* of every cell of the packet to identify where the switch fabric should send it as well as the priority (class) of the packet.
- In the *Cell Header:*
  - Source FlowId:
    Sent in the *Cell Header* to allow the IPE to reassemble the packet. This is simply the identification of the line card (in the least significant 8 bits) and the priority (class) in the most significant two bits. The priority MUST be the same as is specified in the Destination FlowId of this packet.
  - Discard Bit:
    Set by the LC PM to inform the IPE that an error (IP checksum, internal parity) was detected in the packet.
  - EOP Bit:
    Set by the LC PM to indicate the last cell of the packet.
- In the *PIE Header:*
  - Initial PID:

This 3 bit field tells the IPE the type of encapsulation this packet has. The choices are: IPC, Ethernet, PPP, IP, or MPLS.

- Initial Stage:
  This 4 bit field can be used to give additional information to the IPE about the encapsulation of this packet. It specifies which stage in the IPE PI will be the first to inspect the packet.
- Destination PPU (1 bit):
  This bit is set for IP packets whose destination address is equal to one of the IP addresses of the IPE card that packet is being sent to.
- Destination PPUID:
  The PPU identifier of the IPE PPU that the packet is being sent to.
- IPE CID:
  An index into the connection table of the IPE PPU that the packet is being sent to.

Unless MPLS or PPPoE is being used over the Ethernet, each PHY comprises a single *user*. When MPLS or PPPoE is in use, however, each MPLS Label Switched Path (LSP) or PPPoE session represents an additional *user*.

For Ethernet Line Cards, the LC CID is simply the PHYID. The PI DFU control registers can be programmed (by the MPPU) with the LC CID's of up to 4 PHYs. For these PHYs, if the packet contains an IP header, the PI DFU will replace the LC PPUID read from the hardware connection table with a LC PPUID read from a hash table which is indexed by a hash of the source and destination IP addresses of the packet (calculated by the PI DFU). This capability of the PI DFU must be used for 10 Gigabit Ethernet Cards in order to distribute the load over multiple LC PPUs. For 1 Gigabit and smaller PHYs, the PI DFU need not be used. Instead, the PI uses the LC CID as index into the hardware connection table. The PI reads (amongst other things) a LC PPUID which selects the LC PPU that the control information for the packet should be sent to.

Each Ethernet PHY must have a globally unique Ethernet MAC address permanently assigned to it. All Ethernet packets are forwarded to the IPE with their MAC headers intact, and the with the Initial PIDS set to indicate Ethernet, except for MPLS and PPPoE session packets (ethertype = 0x8864).

As packets are inspected by the LC, a determination of priority (one of four classes) is made. This priority is used to determine the two most significant bits of the Destination FlowId when the packet is forwarded to an IPE.

The following is a list of all the different types of top level protocol encapsulation that can be received by the Ethernet line card, along with an explanation of the processing that must take place on the line card for each type of protocol:

1.1.4.1 IP

The LC PPU uses the LC CID (which is really just the PHYID) to index into and read from the software PHY table. From this the LC determines whether this is a *small user* or a *large user*. For *small users*, the *Primary UserId* is also read from the PHY table. This determines where the packet is sent as well as the IPE CID that is sent in the *PIE Header* of the packet.

For *large users*, however, a hash is calculated over the source and destination IP addresses of the packet and used to select either the *Primary UserID* or one of several *Secondary UserIDs*. The selected *UserID* determines where the packet is sent as well as the IPE CID that is sent in the *PIE Header* of the packet.

The packet is forwarded to the IPE with the Ethernet MAC header intact, and the Initial PID is set to indicate an Ethernet packet.

1.1.4.2 PPPoE Session

For PPPoE Session packets, the Ethernet and PPPoE headers are removed, and the PPPoE Session ID (from the PPPoE header) is used to index into a PPPoE session table, from which the UserId (IPE CardId, IPE PPUID and IPE CID) can be retrieved. A unique PPPoE Session table can be maintained for each PHY, and the LC CID can be used to select which session table to use.

If the PPP protocol type is IP, the PPP header is also removed, and the Initial PIDS is set to indicate IP, otherwise, the PPP header is kept, and the Initial PIDS is set to indicate PPP.

1.1.4.3 MPLS

The LC CID only identifies the PHYID that the packet was received on. Therefore, when the LC PI identifies an MPLS packet, the top of stack label must be captured in order to identify the *user*. For each Ethernet PHY, the LC PPU must maintain a table of MPLS LSPs. The LC CID selects which table, and the top of stack label is used to index into the table. For *small* MPLS *users*, the *Primary UserID* that corresponds to the LSP can then be read the table. For *large users*, however, a similar process to the one described above for IP is used. A hash is calculated over the source and destination IP addresses of the packet and used to select either the *Primary UserID* or one of several *Secondary UserIDs*. The selected *UserID* determines where the packet is sent as well as the IPE CID that is sent in the *PIE Header* of the packet.

The Ethernet header is removed before the packet is forwarded to the IPE, and the Initial PID is set to indicate an MPLS packet.

1.1.4.4 Other Ethernet Protocols (ARP, PPPoE Discovery, etc.)

This category includes all Ethernet protocols (ethertypes) other than IP, MPLS, and PPPoE Session.

The LC PPU uses the LC CID (which is really just the PHYID) as an index into a software PHY table. This table provides the *Primary UserId*, which determines where the packet is sent as well as the IPE CID that is sent in the *PIE Header* of the packet. For *large* and *small* Ethernet *users*, these packets are sent to the IPE PPU identified by the *Primary UserId*. No distribution is performed for these packets. Also, the Initial PID is set to indicate an Ethernet packet.

1.2 Line Cards – Egress Side

Line cards perform all the traffic shaping for the system.

1.2.1 ATM Line Card

The following information is received from IPE PPU along with the packet payload:

- In the *CSIX Header:*
  - Destination FlowId:
    Sent in the *CSIX Header* of every cell of the packet to identify where the switch fabric should send it as well as the priority (class) of the packet.
- In the *Cell Header:*
  - Source FlowId:
    Sent in the *Cell Header* to allow the LC to reassemble the packet. This is simply the identification of the IPE card (in the least significant 8 bits) and the priority (class) in the most significant two bits. The priority **MUST** be the same as is specified in the Destination FlowId of this packet.
  - Discard Bit:
    Set by the IPE PM to inform the LC that an error (internal parity) was detected in the packet.
  - EOP Bit:
    Set by the IPE PM to indicate the last cell of the packet.
- In the *PIE Header:*
  - Initial PID:

This 3 bit field tells the LC the type of encapsulation this packet has. The choices are: IPC (for inter-processor communications), IP, Ethernet, PPP, or MPLS

- Initial Stage:
  Always 0.
- Destination PPU (1 bit):
  Always 0.
- Destination PPUID:
  The PPU identifier of the LC PPU that the packet is being sent to.
- LC CID:
  This is an index into the connection table of the LC.

The Destination PPUID selects the LC PPU that will process the packet. The LC CID is used by the LC PPU as an index into a software connection table. This connection table provides the shaping parameters, any additional encapsulation that must be added by the LC, the PHYID, and the ATM VPI/VCI for the packet.

The priority (one of four classes) is based on the two most significant bits of the Source FlowId in the Cell Header. The priority is used by the Traffic Shaper and the Scheduler to determine when to forward the packet to the PHY.

The ATM cell headers and the AAL5 trailer and padding are always added (by the PM) before forwarding the packet to the PHY card.

The following is a list of all the different types of top level protocol encapsulation that can be received from an IPE by the ATM line card, along with an explanation of the processing that must take place on the line card for each type of protocol:

1.2.1.1.1 IP

The desired encapsulation for the packet can be either IP/PPP/PPPoE/Ethernet/ATM, IP/PPP/ATM or IP/ATM. The PPU can determine which it is from the connection table. If the encapsulation should be IP/PPP/PPPoE/Ethernet/ATM, the connection table will provide the necessary information to add the missing headers. If the encapsulation should be IP/PPP/ATM, a PPP header identifying the protocol as IP is added. Also, the entry in the connection table may specify that an LLC header should also be added.

1.2.1.1.2 Ethernet

The connection table may specify that an LLC header should be added to the beginning of the packet. Otherwise the packet is sent as is.

1.2.1.1.3 PPP

The desired encapsulation may be either PPP/PPPoE/Ethernet/ATM or PPP/ATM. The PPU can determine which it is from the connection table. If it is PPP/ATM, the packet is sent as is, otherwise, the connection table will provide the necessary information to add a PPPoE header and an Ethernet Header.

1.2.1.1.4 MPLS

As with all other encapsulations, the proper VPI/VCI is obtained from the connection table.

1.2.2 POS Line Card

The following information is received from IPE PPU along with the packet payload:

- In the *CSIX Header:*
  - Destination FlowId:
    Sent in the *CSIX Header* of every cell of the packet to identify where the switch fabric should send it as well as the priority (class) of the packet.

- In the *Cell Header:*
  - Source FlowId:
    Sent in the *Cell Header* to allow the LC to reassemble the packet. This is simply the identification of the IPE card (in the least significant 8 bits) and the priority (class) in the most significant two bits. The priority **MUST** be the same as is specified in the Destination FlowId of this packet.
  - Discard Bit:
    Set by the IPE PM to inform the LC that an error (internal parity) was detected in the packet.
  - EOP Bit:
    Set by the IPE PM to indicate the last cell of the packet.
- In the *PIE Header:*
  - Initial PID:
    This 3 bit field tells the LC the type of encapsulation this packet has. The choices are: IPC (for inter-processor communications), IP, PPP, or MPLS
  - Initial Stage:
    Always 0.
  - Destination PPU (1 bit):
    Always 0.
  - Destination PPUID:
    The PPU identifier of the LC PPU that the packet is being sent to.
  - LC CID:
    This is an index into the connection table of the LC.

The Destination PPUID selects the LC PPU that will process the packet. The LC CID is used by the LC PPU as an index into a software connection table. This connection table provides the shaping parameters, and the PHYID for the packet.

The following is a list of all the different types of top level protocol encapsulation that can be received by the POS line card, along with an explanation of the processing that must take place on the line card for each type of protocol:

1.2.2.1 PPP
No additional processing is needed.

1.2.2.2 IP
A PPP header identifying the packet as an IP packet is added.

1.2.2.3 MPLS
A PPP header identifying the packet as a MPLS packet is added.

1.2.3 Ethernet Line Card
The following information is received from IPE PPU along with the packet payload:

- In the *CSIX Header:*
  - Destination FlowId:
    Sent in the *CSIX Header* of every cell of the packet to identify where the switch fabric should send it as well as the priority (class) of the packet.
- In the *Cell Header:*
  - Source FlowId:
    Sent in the *Cell Header* to allow the LC to reassemble the packet. This is simply the identification of the IPE card (in the least significant 8 bits) and the priority (class) in the most significant two bits. The priority **MUST** be the same as is specified in the Destination FlowId of this packet.
  - Discard Bit:

Set by the IPE PM to inform the LC that an error (internal parity) was detected in the packet.

- EOP Bit:
  Set by the IPE PM to indicate the last cell of the packet.

- In the *PIE Header:*
  - Initial PID:
    This 3 bit field tells the LC the type of encapsulation this packet has. The choices are: IPC (for inter-processor communications), Ethernet, PPP, IP, or MPLS
  - Initial Stage:
    Always 0.
  - Destination PPU (1 bit):
    Always 0.
  - Destination PPUID:
    The PPU identifier of the LC PPU that the packet is being sent to.
  - LC CID:
    This is an index into the connection table of the LC.

The Destination PPUID selects the LC PPU that will process the packet. The LC CID is used by the LC PPU as an index into a software connection table. This connection table provides the shaping parameters, and the PHYID for the packet.

The following is a list of all the different types of top level protocol encapsulation that can be received by the Ethernet line card, along with an explanation of the processing that must take place on the line card for each type of protocol:

1.2.3.1 Ethernet

No additional processing is needed. All IP/Ethernet are sent using this type because the IPE, not the LC implements ARP, and therefore adds the Ethernet header to all IP packets before sending them to the LC.

1.2.3.2 PPP

The desired encapsulation is PPP/PPPoE/Ethernet. The connection table provides the necessary information to add a PPPoE header and an Ethernet header.

1.2.3.3 IP

The desired encapsulation is IP/PPP/PPPoE/Ethernet. A PPP header indicating an IP packet is added. The connection table then provides the necessary information to add a PPPoE header and an Ethernet header.

1.2.3.4 MPLS

The connection table provides the information needed to add an Ethernet header (the destination MAC address is all that is required from the connection table).

1.3 IPE Card

1.3.1 IPE Ingress Protocols

All packets received by an IPE card from the Line Cards (or from other IPEs) will be of one of the following types. The Initial PID field in the *PIE Header* will identify which one of these types each packet corresponds to. If there are more than 8 such types, the Initial Stage field in the *PIE Header* can be used to select a different stage to begin inspection, each of which allows 8 additional protocols to be identified by the *Initial PID* field.

The IPE CID and PPUID in the PIE Header of the received packet combine with the FlowId to give the UserId. Only the least significant 18 of the 20 bits of the IPE CID are used.

1.3.1.1 IPC

These packets are used for inter-processor communication within the system. The PI should be programmed to capture these packets to a PPU (as specified in the Destination PPU field in the PIE Header) in their entirety.

1.3.1.2 IP

These packets consist of only an IP packet. That is, an IP header, followed by an IP payload (which might include TCP, UDP, ICMP, etc.). This category does not include IP packets received over MPLS or over Ethernet.

These packets can come from either a POS LC, an ATM LC, or an Ethernet LC. The possible encapsulations that could result in such a packet are: IP/ATM, IP/PPP/ATM, IP/PPP/SONET, IP/PPP/PPPoE/Ethernet, and IP/PPP/PPPoE/Ethernet/ATM.

The IPE CID uniquely identifies the PPPoE Session ID, or the ATM Virtual Circuit that the packet was received on as well as the PHY/LC that it was received on. In the case of IP/PPP/SONET, the IPE CID will identify only the PHY/LC that the packet was received on, that is, it will be constant for all IP/PPP/SONET packets received from a particular PHY/LC.

1.3.1.3 PPP

This category consists of all PPP packets received whose PPP protocol type was not IP or MPLS. These packets can come from a POS LC, an ATM LC, or an Ethernet LC. For those PPP sessions that will be tunneled using L2TP, the IPE must add a new PPP header to the IP/PPP and MPLS/PPP packets, since for those protocols, the PPP header will have been removed by the Line Card.

The possible encapsulations that could result in such a packet are: PPP/SONET, PPP/PPPoE/Ethernet, PPP/ATM, and PPP/PPPoE/Ethernet/ATM.

The IPE CID uniquely identifies the PPPoE Session ID, or the ATM Virtual Circuit that the packet was received on as well as the PHY/LC that it was received on. In the case of PPP/SONET, the IPE CID will identify only the PHY/LC that the packet was received on, that is, it will be constant for all PPP/SONET packets received from a particular PHY/LC.

1.3.1.4 Ethernet

This category consists of all Native Ethernet or Ethernet/ATM packets received except for MPLS (ethertype = 0x????) and PPPoE data packets (ethertype = 0x8864). This category also includes packets whose destination MAC address is not equal to the MAC address of the PHY on which the packet was received (broadcast and multicast packets, and unicast packets if in promiscuous mode).

The possible encapsulations that could result in such a packet are: ARP/Ethernet, IP/Ethernet, PPPoE Discovery/Ethernet, ARP/Ethernet/ATM, IP/Ethernet/ATM, and PPPoE Discovery/Ethernet/ATM.

For Ethernet/ATM, the IPE CID uniquely identifies the ATM Virtual Circuit that the packet was received on as well as the PHY/LC that it was received on. In the case of Native Ethernet, the IPE CID will identify only the PHY/LC that the packet was received on, that is, it will be constant for all packets received from a particular PHY/LC.

1.3.1.5 MPLS

This category consists of packets which begin with an MPLS label stack. These can come from a POS LC, an ATM LC or an Ethernet LC.

The possible encapsulations that could result in such a packet are: MPLS/PPP/SONET, MPLS/Ethernet, and MPLS/ATM. In the case of MPLS/ATM, the Line Card will have replaced the

top of stack shim label with the real label because the real label was encoded as the ATM VPI/VCI in the packet received from the network.

The following encapsulations are NOT supported: MPLS/PPP/PPPoE/Ethernet, MPLS/PPP/ATM, MPLS/PPP/PPPoE/Ethernet/ATM, and MPLS/Ethernet/ATM.

The IPE CID uniquely identifies the same as the top of stack incoming top of stack MPLS label, as well as the PHY/LC that it was received on. In the case of MPLS/ATM the top of stack label has a one to one correspondence with the ATM Virtual Circuit that the packet was received on.

1.3.2 IPE Ingress Protocol Decoding

The following table shows the first two layers of protocols that must be identified by the PI on the IPE for each packet that passes through it.

IP	
Ethernet	IP
	ARP
	PPPoE Discovery
MPLS	IP
PPP	Control Protocols

1.3.2.1 IP Packets

All IP packets received by the IPE, whether still encapsulated with Ethernet, with MPLS, or without encapsulation, will fall into one of two categories: those for which the destination IP address is equal to one of the addresses of the IPE, and those for which it isn't. In the case of the latter, the packet must be forwarded or discarded by the PPU. But for the former, it must be determined whether or not the packet can be processed entirely by the PPU, or whether it must be sent to the MPPU for further processing. If it must be sent to the MPPU, it must be captured in its entirety.

All IP packets received, regardless of their encapsulation, must have their destination IP address captured and examined. All routing table searches are performed by the IPE cards. If the destination address is one of the system's IP addresses, but not one of the IPE card's addresses, the packet must be forwarded with Destination PPU bit set.

1.3.2.2 L2TP Tunnels

Each L2TP tunnel is handled entirely by a particular IPE card. Each session within the tunnel must be handled entirely by a particular PPU. This requirement comes primarily from the need to support sequence numbers on the data sessions:

> *RFC-2661: "Each peer maintains separate sequence numbers for the control connection and each individual data session within a tunnel."*

Therefore, *large* PPP *users* cannot be tunneled. All L2TP control packets are forwarded to and processed by the MPPU of the IPE card.

1.3.2.2.1 L2TP Access Concentrator (LAC)

Any PPP *user* can be selected for L2TP tunneling by the IPE MPPU. If a user is selected for tunneling, then the PPU receiving PPP packets from that *user* must encapsulate those packets, first with an L2TP header, then a UDP header, and finally an IP header. The IP header's destination address will be that of the configured LNS, and the source address will be one of the IP addresses of

IPE. The resulting IP packet can then be forwarded using the standard IP forwarding procedure to the appropriate Line Card for transmission. It should be evident that tunneled PPP *users* on different IPE cards will be placed in separate tunnels even if being tunneled to the same destination LNS.

IP packets received from the LNS will be sent by the receiving Line Card to the IPE PPU associated with the ingress interface (*user*). This PPU may well be on a different IPE card than the one handling the tunnel. This is easily determined from the destination IP address of the packet. In this case, the PPU receiving the packet from the Line Card must forward the packet to the IPE card handling the tunnel. In addition, the L2TP Session ID can be used to identify which PPU on that IPE card should receive the packet (this PPUID must be sent in the PIE Header so that the receiving PI will know which PPU should receive the packet). This is done by always encoding the PPUID of the PPU handling a particular session in the most significant four bits of the L2TP Session ID.

> *RFC-2661: "Since L2TP sessions are named by identifiers that have local significance only. That is, the same session will be given different Session Ids by each end of the session. Session ID in each message is that of the intended recipient, not the sender."*

The PPU to which the packet is sent to can in turn can de-encapsulate the PPP packet and forward it to the PPP *user* identified by the L2TP session ID.

1.3.2.2.2 L2TP Network Server (LNS)

When functioning as an LNS, L2TP packets received from the LAC will be forwarded, either by a Line Card or another IPE, to the IPE handling the tunnel. This is because the destination IP address of the packet will be equal to one of the IP addresses of the IPE handling the tunnel. Within that IPE, the PPU that should process the L2TP session is identified using the most significant four bits of the L2TP Session ID. The PPU will de-encapsulate the PPP packet, then process the PPP packet as if it was received from a PPP *user*. From this point on, the processing is the same as for a "real" PPP *user*.

In the other direction, packets which, when their destination IP address is looked up in the routing table, yield a destination PPP *user* that is associated with a L2TP tunnel instead of with a Line Card, must be sent to the IPE PPU handling the PPP *user*. This is because of the sequence number requirement of L2TP mentioned above. Once received this PPU, the packet must have a PPP header added, as is the case with a normal PPP user. Then, instead of forwarding the packet to a Line Card, a L2TP header is added, followed by a UDP header and an IP header. The IP destination address is that of the LAC at the other end of the tunnel. The resulting IP packet can then be forwarded using the standard IP forwarding procedure to the appropriate Line Card for transmission.

1.3.2.3 IPSec Tunnels

Each IPSec Security Association (SA) is handled entirely by a particular IPE PPU. As defined in RFC-2401, a Security Association is a unidirectional, "simplex" connection that provides security services to the traffic carried by it.

1.3.2.3.1 Inbound IPsec processing

- Plain packets

Every PPU must have a copy of the SPD for every *user* from which it receives packets. In other words, for every *UserID* (*Primary* or *Secondary*) that points to a particular IPE PPU, the PPU must have a pointer to an SPD. If a *user's* traffic is split among multiple PPUs (i.e.: a *large user*), then they should have identical SPDs configured for the *user*, and each will create its own set of Security Associations for its share of the *user*'s traffic. Every packet received must be processed using the SPD of the *user* the packet is received from.

- Tunneled packets

The SPI is the field in the IPSec header that, along with the destination IP address, identifies the SA. Traffic from a *small user* will always be directed by receiving Line Card to a particular PPU. This PPU uses the SPI to identify the SA, and thus has access to the information it needs to decapsulate the packet. For large users, however, the Line Card must detect IPsec packets whose IP destination address is one of the addresses that belongs to the IPE card identified by the *user's Primary UserId*. Rather than select a *UserId* (*primary* or *secondary*) based on the hash of the source and destination IP addresses of the packet, the LC must use the SPI in the IPsec header to select the *UserId*, and thus the IPE PPU, to send the packet to. In order to accomplish this, the most significant 4 bits of an SPI always contain the PPUID identifying the PPU that is handling the SA identified by that SPI.

1.3.2.3.2 Outbound IPsec processing

Since IPSec performs tunneling at Layer 3, entire *users* don't get tunneled. Rather, each packet about to be sent to a *user* is individually examined using the Security Policy Database (SPD) associated with that *user*, from which a pointer to a SA (in the SAD also associated with the *user*) is obtained.

The difficulty with outbound processing is that, as discussed earlier, the configuration information (and thus the SPD) associated with the egress *user* is not readily available. The information must be requested from the PPU identified by the *Primary UserId* and stored in a cache. Each PPU sending to a *user* will thus create its own set of Security Associations.

1.3.3 IPE Packet Forwarding and Egress Processing

The IPE card PPUs performs routing table searches for all packets that need forwarding. The global Forwarding Information Base (FIB) is distributed to every PPU in the system, and contains IP unicast and multicast routing tables in a form that facilitates longest matching prefix searches (i.e.: Patricia tries), as well as tables required for MPLS label based forwarding.

One of the results of every routing table lookup is the *Primary UserId* identifying the layer 2 interface by which the packet should be transmitted. It is important to note that the *Primary UserId* is not the same as the *LCUserId*, and does not directly give the CardId of the Line Card where the packet should be forwarded. Rather, the *Primary UserId* identifies the IPE PPU that maintains the configuration and state information for the *user*.

This presents a complication because the IPE that is trying to forward the packet needs the information stored on the IPE PPU identified by the *Primary UserId*. Rather than simply forward the packet to the other IPE for egress processing, which would result in additional latency and switch fabric bandwidth utilization for every packet, it sends a message to the PPU identified by the *Primary UserId*, requesting a copy of the *user's* configuration. This information is kept in a user configuration cache and is used for all subsequent packets directed to the same *user*. All counters and statistics that need to be maintained for each user must also be maintained for each cached user, and must also be periodically sent to the PPU identified by the *Primary UserId*.

This process makes it difficult to implement such functionality as per-user traffic shaping in the IPE PPU, because the processing would need to be distributed among a potentially large number of processors. Therefore, traffic shaping is to be implemented strictly on the Line Card using the egress PPUs.

One of the fields that is acquired and cached as part of the user configuration information is the *LCUserId*. This field contains the CardId of the Line Card that the packet must be forwarded to, as well as the PPUID and CID that should be sent in the PIE header of the packet to that Line Card.

What is claimed is:

1. A packet processing circuit comprising:

a packet inspector for examining a stream of cells to determine control information for packets represented thereby;

at least one buffer access controller connected to said packet inspector for storing at least a portion of data cells received from said packet inspector, and for processing control information received from said packet inspector to produce additional control information; and

a packet manager connected to said buffer access controller for receiving control information therefrom for use in formatting packets corresponding to said control information.

2. The circuit of claim 1 wherein said packet manager is configured for using the control information received from said buffer access controller to reassemble said corresponding packets.

3. The circuit of claim 2 wherein said packet manager is connected to said packet inspector for coordinating the dequeuing of data cells representing said corresponding packets from said buffer access controller.

4. The circuit of claim 1 further comprising a cell buffer associated with said buffer access controller for storing said data cells.

5. The circuit of claim 4 further comprising at least one protocol processing unit associated with said buffer access controller for processing said control information received from said packet inspector.

6. The circuit of claim 5 wherein said protocol processing unit comprises at least one general purpose processor unit.

7. The circuit of claim 5 further comprising an additional buffer access controller connected to said packet inspector, wherein said buffer access controllers are configured for storing different portions of data cells received from said packet inspector.

8. The circuit of claim 7 further comprising a protocol processing unit associated with said additional buffer access controller, and wherein said buffer access controllers are each configured for determining whether to forward certain control information received from said packet inspector to its associated protocol processing unit for processing.

9. The circuit of claim 8 further comprising a master processing unit connected to said protocol processing units for providing said protocol processing units with configuration data.

10. The circuit of claim 9 further comprising a switch, wherein said master processing unit and said protocol processing units are interconnected to one another through said switch.

11. The circuit of claim 7 wherein each buffer access controller has at least two protocol processing units associated therewith.

12. A mid-network server comprising:

an input for receiving a packet delivered thereto;

a line module connected to said input for receiving said packet;

a plurality of processing modules for performing mid-network processing functions; and

a switch fabric connected to said line module and said processing modules for delivering packets therebetween, wherein said processing modules are at

least substantially identical to one another and independently programmable.

13. The server of claim 12 further comprising an additional line module, wherein said line modules are at least substantially identical to one another and independently programmable.

14. The server of claim 12 wherein said processing modules are each configured to support a plurality of packet types, and each line module is configured for formatting a packet into one of said types prior to sending said packet through said switch fabric to one of said processing modules.

15. The server of claim 12 wherein said line module and said processing modules each comprise a packet inspector, a packet manager, and at least one buffer access controller.

16. The server of claim 15 wherein said line module and said processing modules each comprise a plurality of buffer access controllers interconnected with said packet inspector and said packet manager.

17. The server of claim 16 wherein each of said buffer access controllers have at least one protocol processing unit associated therewith.

18. The server of claim 17 wherein each protocol processing unit is in communication with at least one other protocol processing unit on the same module.

19. The server of claim 18 wherein said line module and said processing modules each comprise a master protocol processing unit for controlling the protocol processing units on that module.

20. The server of claim 19 wherein said line module and said processing modules each comprise an Ethernet switch for interconnecting the master protocol processing

unit with said other protocol processing units on that module.

21. A packet server comprising:

an input for receiving a packet delivered thereto;

a line module connected to said input for receiving said packet;

a plurality of processing modules for performing packet routing functions; and

a switch fabric connected to said line module and said processing modules for delivering packets therebetween, wherein said line module is configurable to send said packet to any one of said processing modules through said switch fabric, and said processing modules are each configurable to perform said routing functions for said packet if said packet is sent thereto by said line module.

22. The server of claim 21 wherein said line module supports a plurality of user interfaces and is configured to send said packet to one of said processing modules according to the user interface through which said packet arrives at said server.

23. The server of claim 22 wherein each processing module includes a plurality of processing units, and said line module is configured to send said packet to one of said processing units of one of said processing modules according to the user interface through which said packet arrives at said server.

24. The server of claim 23 wherein said processing modules are each configured to support a plurality of packet types, and said line module is configured for formatting said packet into one of said types prior to sending said packet through said switch fabric to one of said processing modules.

25. The server of claim 21 wherein said line module is connected to said input through a phy module.

26. The server of claim 21 wherein said line module and said processing modules each include a plurality of general purpose processing units.

27. The server of claim 21 wherein said line module and said processing modules can be programmed to support any type of transmission protocol.

28. A packet server comprising:

a plurality of line modules for receiving packets delivered to said server over a physical connection;

at least one processing module for performing packet routing functions; and

a switch fabric connected to said line modules and said processing module for delivering packets therebetween, wherein each line module is configured to format a packet into one of a plurality of types prior to sending said packet through said switch fabric to said processing module, and said processing module is configured to support each of said packet types.

29. The server of claim 28 wherein said processing module includes a plurality of processing units.

30. The server of claim 29 wherein each line card supports a plurality of users and is configured to assign each user to one of the processing units of said processing module.

31. The server of claim 30 wherein at least one of the processing units of said processing module is assigned to a first user supported by a first one of said line modules and a second user supported by a second one of said line modules.

32. A method for processing packets within a server, said method comprising the steps of:

converting a packet input to said server into a stream of fixed length cells;

processing said stream of fixed length cells using a line module to format said packet into one of a plurality of protocol types; and

sending said formatted packet to a processing module configured to support each of said plurality of protocol types.

33. The method of claim 32 wherein said processing step includes reassembling said packet.

34. The method of claim 33 wherein said processing step further includes examining said cell stream to obtain control information for said packet.

35. The method of claim 34 wherein said control information includes information identifying a particular processing module for further processing said packet.

36. The method of claim 34 wherein said processing step further includes processing said control information to produce additional control information for use in reassembling and formatting said packet.

37. The method of claim 36 wherein said processing step further includes identifying a particular processing module to which said packet should be sent.

38. The method of claim 37 wherein the sending step includes sending said packet to said particular processing module identified in said processing step.

39. The method of claim 37 wherein said identifying step includes identifying a particular protocol processing unit on said particular processing module for processing control information corresponding to said packet.

40. The method of claim 33 further comprising the step of performing mid-network processing functions on said sent packet using said processing module.

41. The method of claim 40 wherein the step of performing mid-network processing functions includes formatting said packet for its destination interface.

42. The method of claim 41 further comprising the step of sending the packet formatted by said processing module to a line module corresponding to said destination interface.

43. The method of claim 32 wherein said sending step includes sending said packet through a switch fabric.

44. The method of claim 32 wherein said input packet is a packet represented by a plurality of fixed length cells.

45. The method of claim 44 wherein the converting step includes modifying the length of said input cells.

46. A method for processing packets within a server, said method comprising the steps of:

converting a packet input to said server into a stream of fixed length cells;

processing said stream of fixed length cells using a line module to format said packet into one of a plurality of protocol types; and

sending said formatted packet to another line module configured to support each of said plurality of protocol types.

47. A method for processing packets within a server, said method comprising the steps of:

converting a packet input to said server into a stream of fixed length cells;

processing said stream of fixed length cells using a line module to format said packet into one of a plurality of protocol types;

sending said formatted packet to a processing module configured to support each of said plurality of protocol types; and

processing said stream of fixed length cells in said processing module;

reformatting said formatted packet into one of a plurality of protocol types; and

sending said reformatted packet to another processing module.

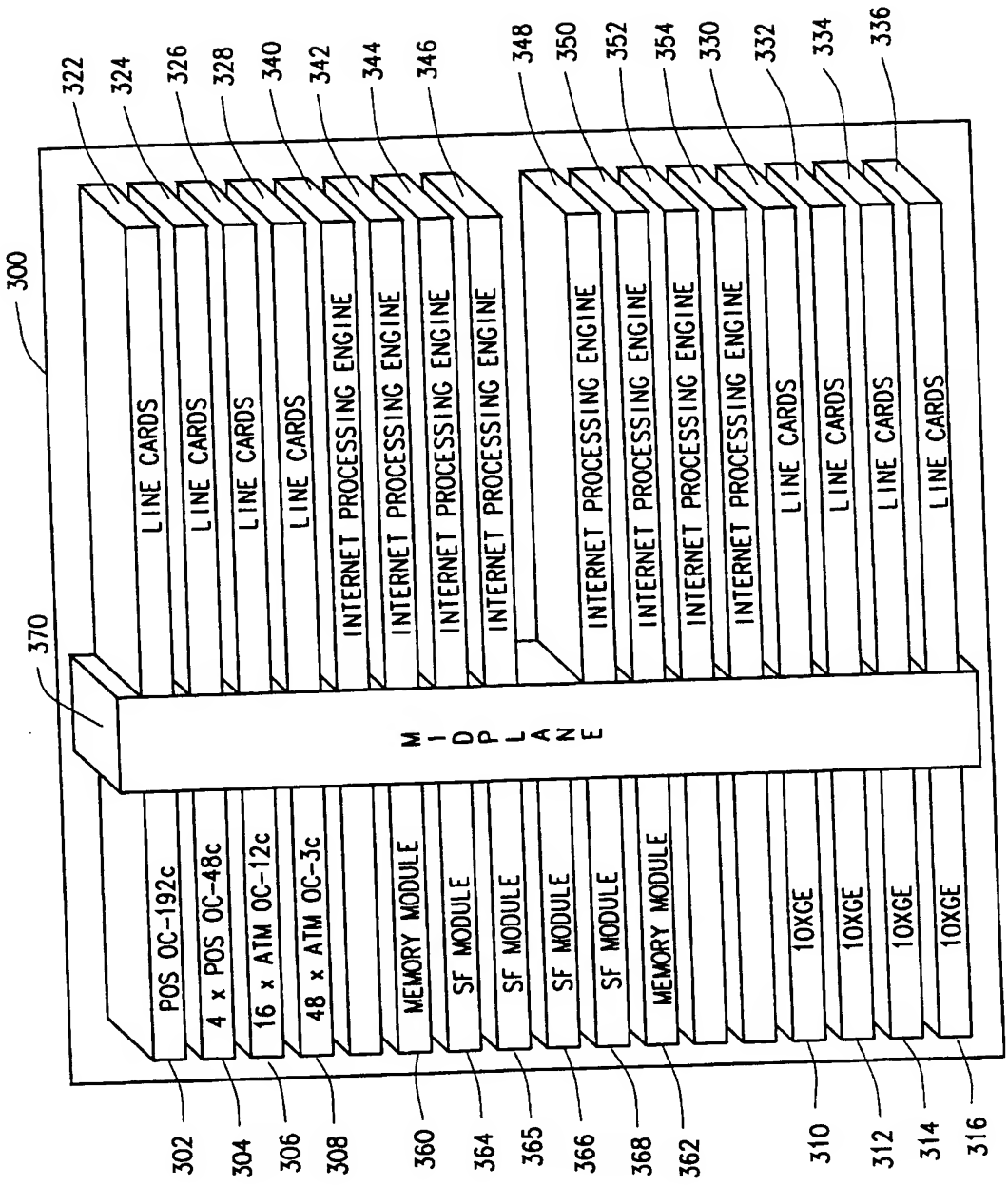

300
370
322
324
326
328
340
342
344
346
348
350
352
354
330
332
334
336
LINE CARDS
LINE CARDS
LINE CARDS
LINE CARDS
INTERNET PROCESSING ENGINE
INTERNET PROCESSING ENGINE
INTERNET PROCESSING ENGINE
INTERNET PROCESSING ENGINE
INTERNET PROCESSING ENGINE
INTERNET PROCESSING ENGINE
INTERNET PROCESSING ENGINE
INTERNET PROCESSING ENGINE
LINE CARDS
LINE CARDS
LINE CARDS
LINE CARDS
MIDPLANE
POS OC-192c
4 x POS OC-48c
16 x ATM OC-12c
48 x ATM OC-3c
MEMORY MODULE
SF MODULE
SF MODULE
SF MODULE
SF MODULE
SF MODULE
MEMORY MODULE
10XGE
10XGE
10XGE
10XGE
302
304
306
308
360
364
365
366
368
362
310
312
314
316

FIG. 1

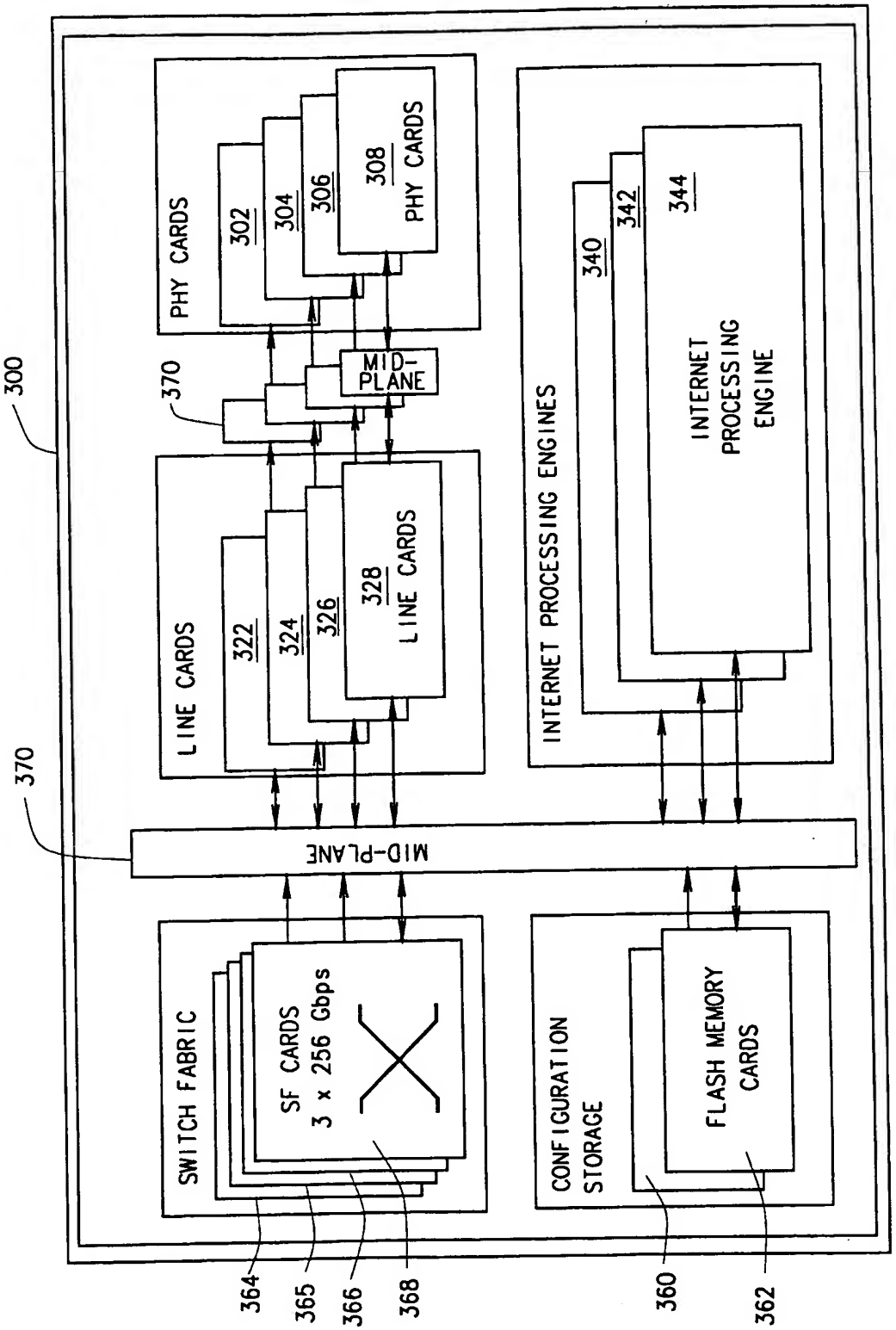

300
370
370
PHY CARDS
302
304
306
308
PHY CARDS
MID PLANE
LINE CARDS
322
324
326
328
LINE CARDS
INTERNET PROCESSING ENGINES
340
342
344
INTERNET PROCESSING ENGINE
MID-PLANE
SWITCH FABRIC
SF CARDS
3 x 256 Gbps
CONFIGURATION STORAGE
FLASH MEMORY CARDS
364
365
366
368
360
362

FIG. 2

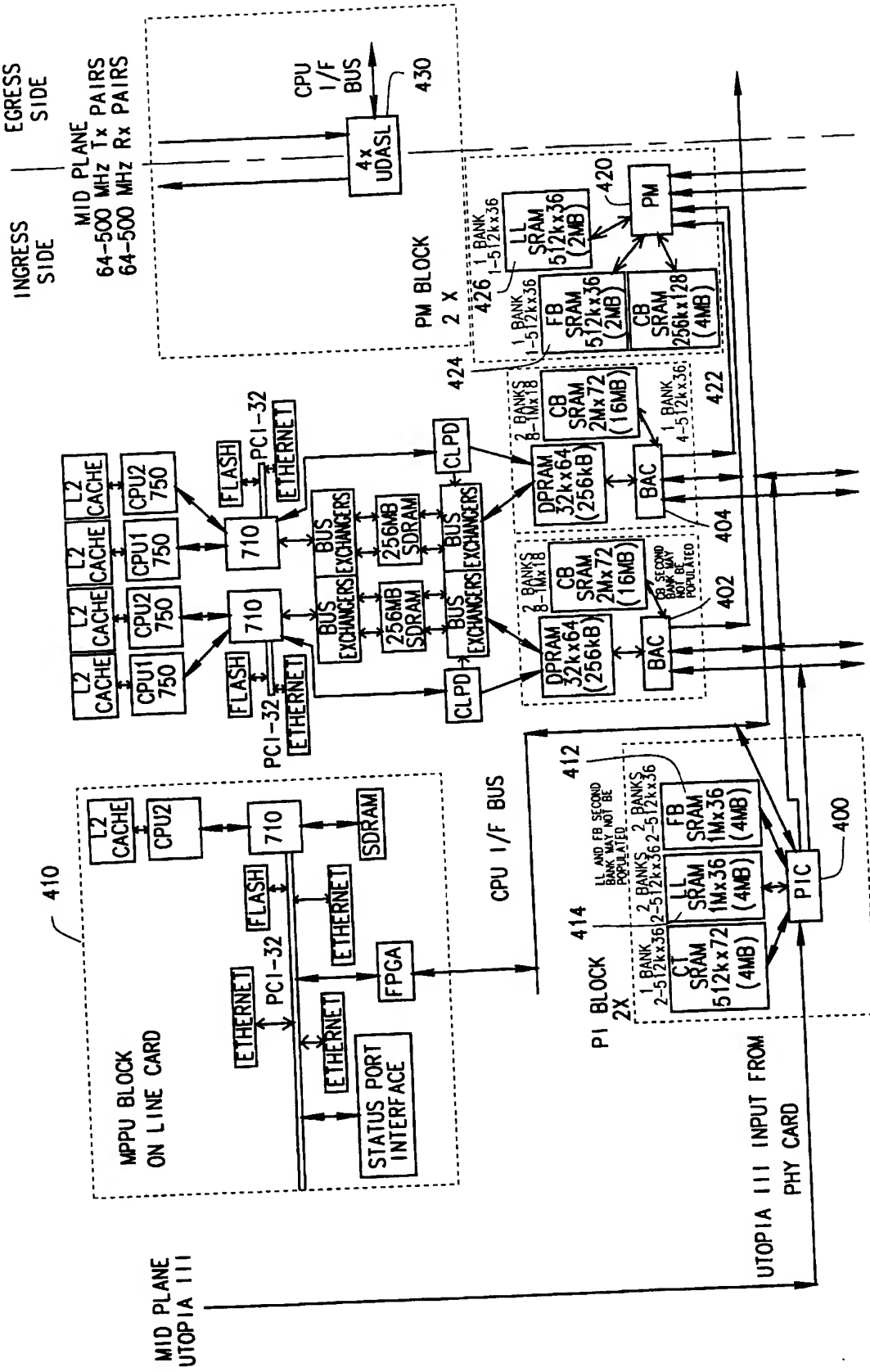
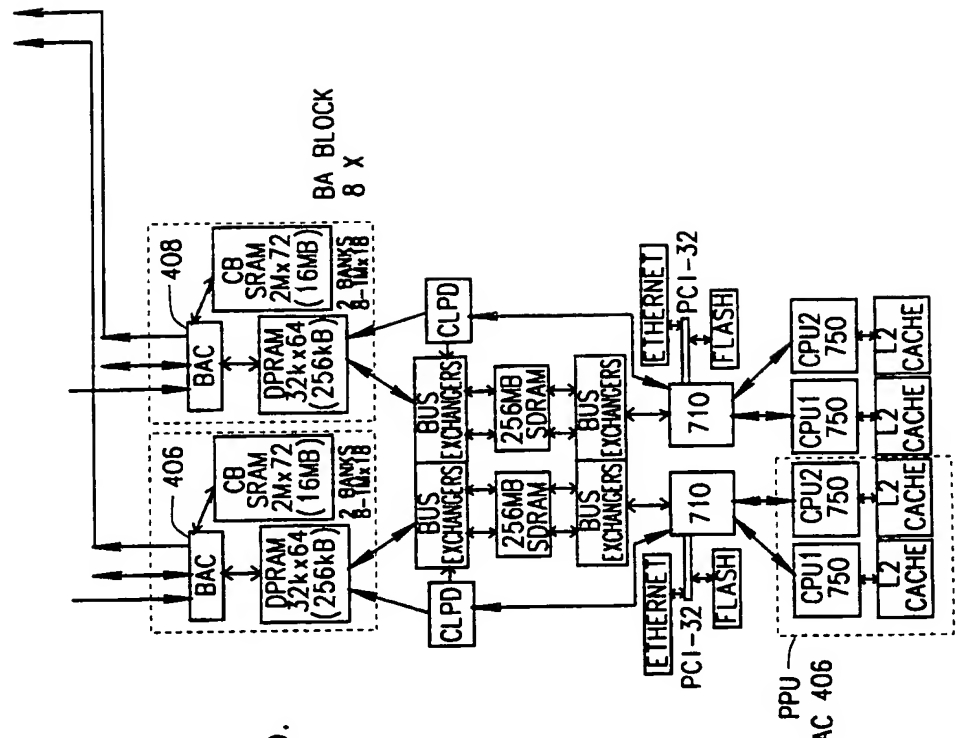

EGRESS SIDE
INGRESS SIDE
MID PLANE
64-500 MHz Tx PAIRS
64-500 MHz Rx PAIRS
CPU I/F BUS
430
4x UDASL
PM BLOCK 2 X
426
1 BANK 1-512kx36
LL SRAM 512kx36 (2MB)
FB SRAM 512kx36 (2MB)
CB SRAM 256kx128 (4MB)
PM
420
424
422
2 BANKS 8-1Mx18
CB SRAM 2Mx72 (16MB)
DPRAM 32kx64 (256kB)
BAC
1 BANK 4-512kx36
404
402
FB SECOND BANK MAY NOT BE POPULATED
CLPD
BUS EXCHANGERS
256MB SDRAM
710
CPU1 750
CPU2 750
L2 CACHE
FLASH
PCI-32
ETHERNET
410
MPPU BLOCK ON LINE CARD
SDRAM
FPGA
STATUS PORT INTERFACE
CPU I/F BUS
412
P1 BLOCK 2X
414
LL AND FB SECOND BANK MAY NOT BE POPULATED
2 BANKS 2-512kx36
FB SRAM 1Mx36 (4MB)
LL SRAM 1Mx36 (4MB)
CT SRAM 512kx72 (4MB)
P1C
400
MID PLANE UTOPIA III
UTOPIA III INPUT FROM PHY CARD

FIG. 3A



NOTE: BA CB SECOND BANK MAY NOT BE POPULATED (PROBABLY WON'T DUE TO POWER DISSIPATION.)

NOTE: P1LL AND FB SECOND BANK MAY NOT BE POPULATED.

NOTE: ALL SRAM S1 133MHz ZBT SRAM EXCEPT PM CB IS 200 MHz DDR SSRAM.

NOTE 1: JTAG BUS GOES TO ALL MICROPROCESSORS, ASICS, AND FPGA'S.

FIG. 3B

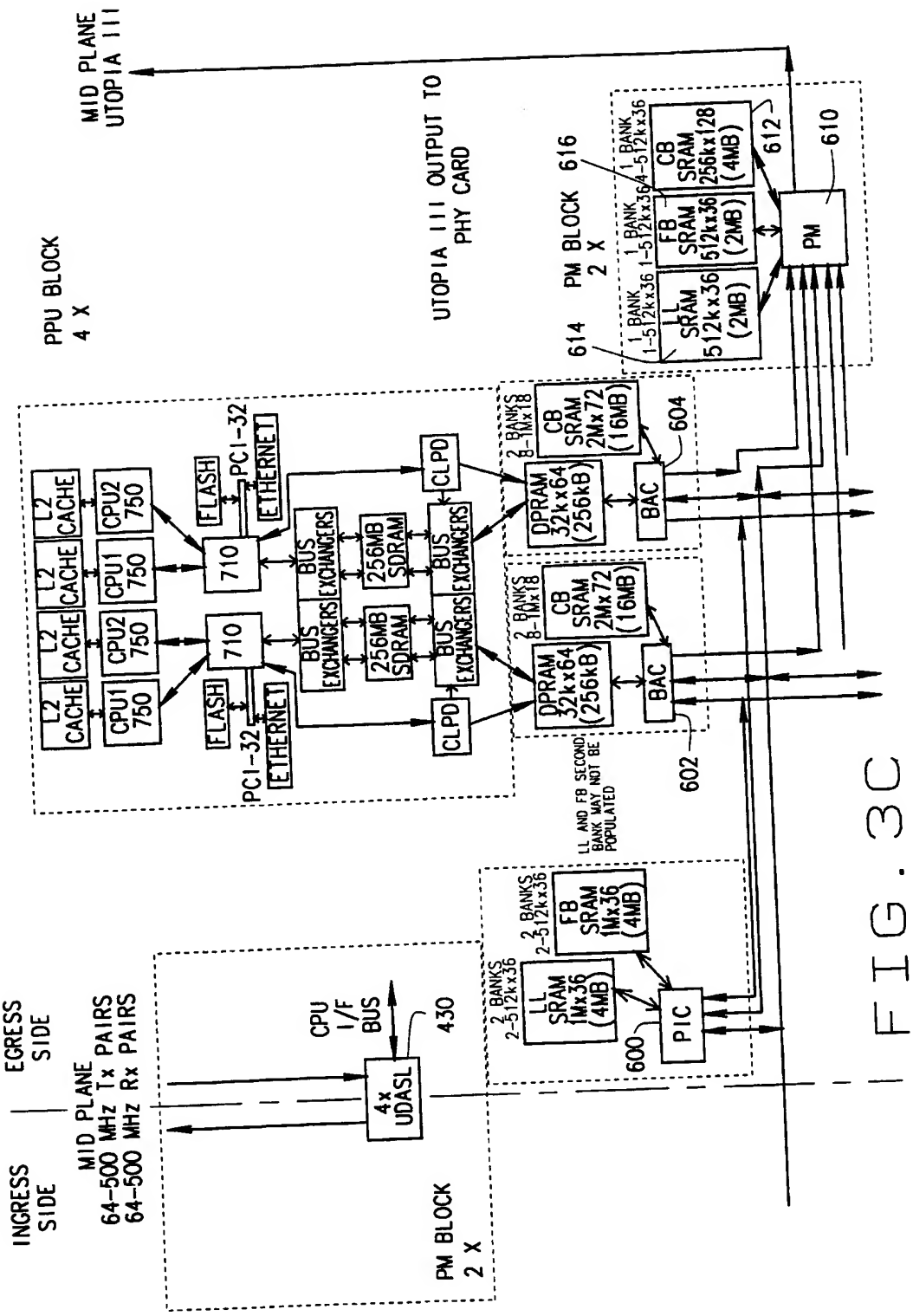

MID PLANE UTOPIA III
UTOPIA III OUTPUT TO PHY CARD
PPU BLOCK 4 X
PM BLOCK 2 X
614
PM BLOCK 616 2 X
L2 CACHE
CPU1 750
CPU2 750
710
FLASH
PCI-32
ETHERNET
BUS EXCHANGERS
256MB SDRAM
CLPD
DPRAM 32kx64 (256kB)
2 BANKS 8-1Mx18
CB SRAM 2Mx72 (16MB)
BAC
604
602
1 BANK 1-512kx36
LL SRAM 512kx36 (2MB)
FB SRAM 512kx36 (2MB)
CB SRAM 256kx128 (4MB)
PM
612
610
LL AND FB SECOND BANK MAY NOT BE POPULATED
2 BANKS 2-512kx36
LL SRAM 1Mx36 (4MB)
FB SRAM 1Mx36 (4MB)
PIC
600
CPU I/F BUS
430
4x UDASL
EGRESS SIDE
INGRESS SIDE
MID PLANE
64-500 MHz Tx PAIRS
64-500 MHz Rx PAIRS

FIG. 3C

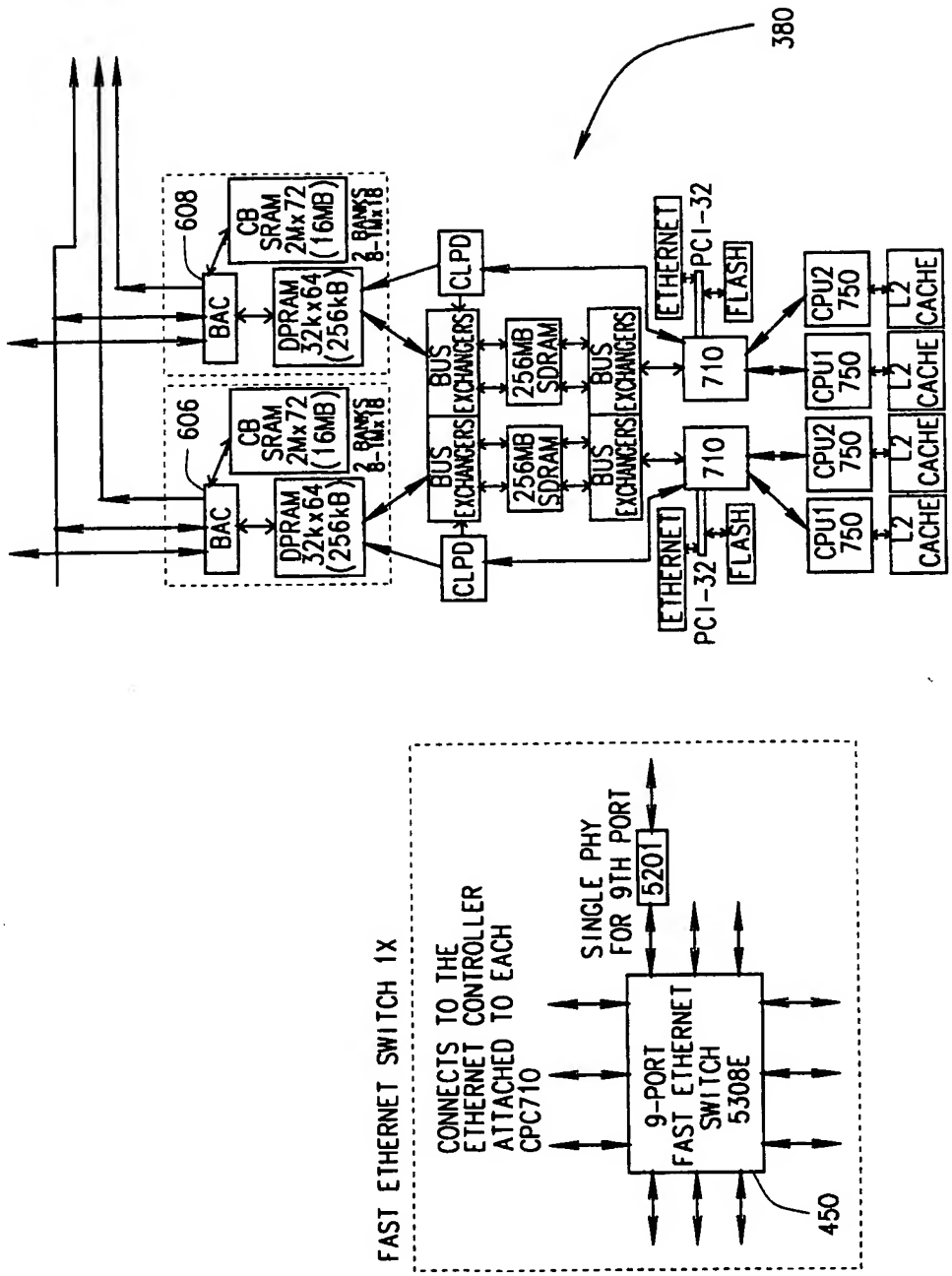

380
608
606
CB SRAM 2Mx72 (16MB)
2 BANKS 8-18x18
DPRAM 32kx64 (256kB)
BAC
CLPD
BUS EXCHANGERS
256MB SDRAM
710
ETHERNET
PCI-32
FLASH
CPU2 750
CPU1 750
L2 CACHE
FAST ETHERNET SWITCH 1X
CONNECTS TO THE ETHERNET CONTROLLER ATTACHED TO EACH CPC710
SINGLE PHY FOR 9TH PORT
5201
9-PORT FAST ETHERNET SWITCH 5308E
450

FIG. 3D

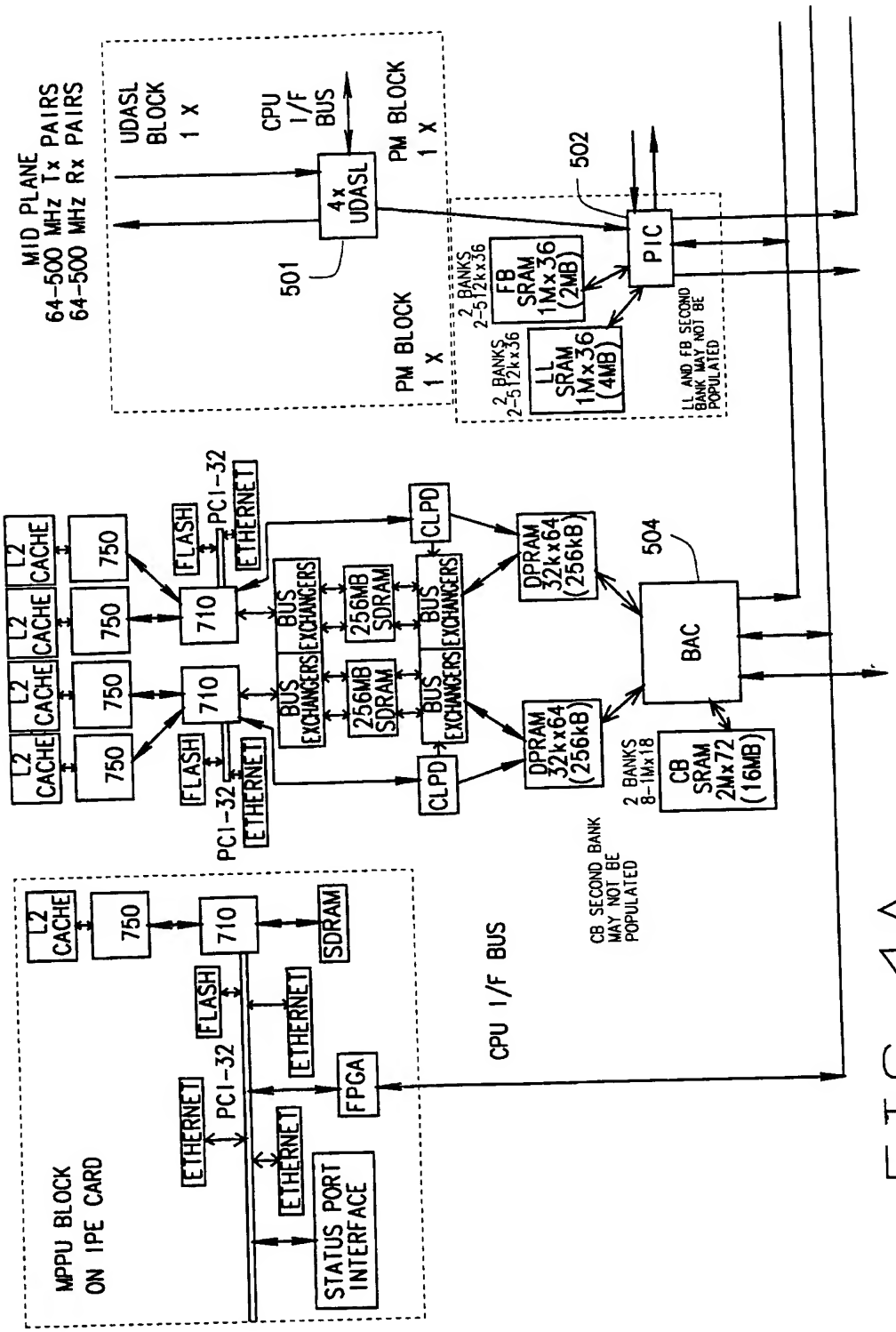
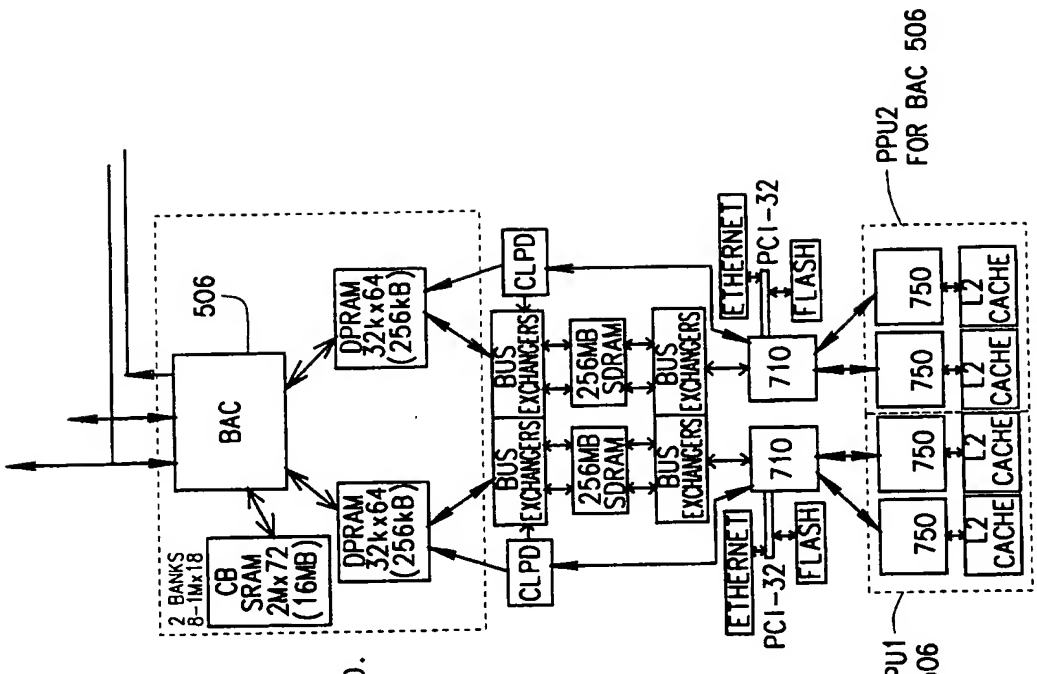

MID PLANE
64-500 MHz Tx PAIRS
64-500 MHz Rx PAIRS
UDASL BLOCK 1 X
CPU I/F BUS
PM BLOCK 1 X
4x UDASL
501
502
PIC
FB SRAM 1Mx36 (2MB)
2 BANKS 2-512kx36
LL SRAM 1Mx36 (4MB)
LL AND FB SECOND BANK MAY NOT BE POPULATED
L2 CACHE
750
710
FLASH
PCI-32
ETHERNET
BUS EXCHANGERS
256MB SDRAM
CLPD
DPRAM 32kx64 (256kB)
504
BAC
2 BANKS 8-1Mx18
CB SRAM 2Mx72 (16MB)
CB SECOND BANK MAY NOT BE POPULATED
CPU I/F BUS
MPPU BLOCK ON IPE CARD
SDRAM
FPGA
STATUS PORT INTERFACE

FIG. 4A



NOTE: BA CB SECOND BANK MAY NOT BE POPULATED (PROBABLY WON'T DUE TO POWER DISSIPATION).

NOTE: PI LL AND FB SECOND BANK MAY NOT BE POPULATED.

NOTE: ALLSRAM IS 133MHz ZBT SRAM EXCEPT PM CB IS 200 MHz DDR SSRAM.

FIG. 4B

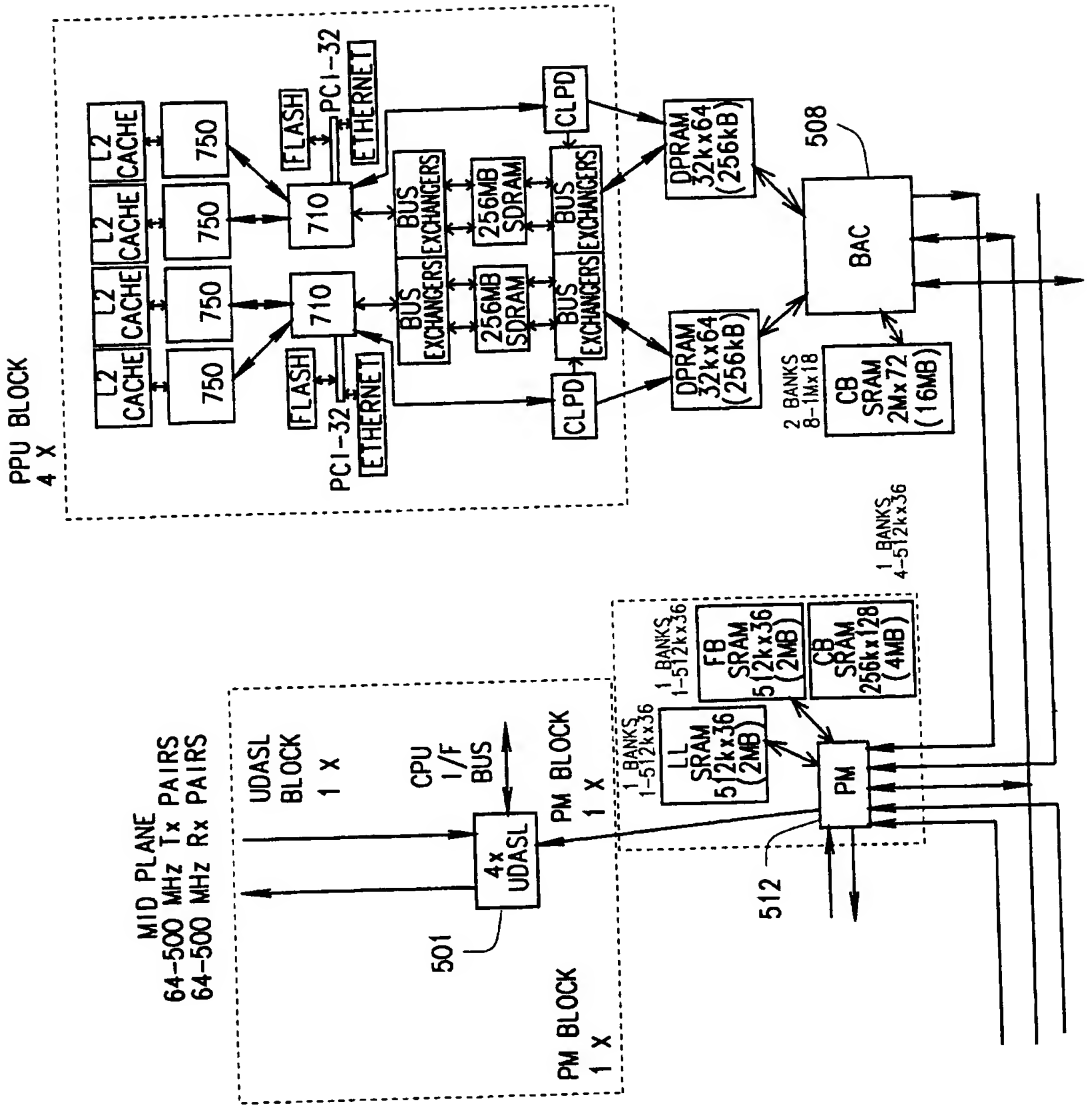

PPU BLOCK
4 X
L2 CACHE
750
710
FLASH
PCI-32
ETHERNET
BUS EXCHANGERS
256MB SDRAM
CLPD
DPRAM 32kx64 (256kB)
508
BAC
2 BANKS 8-1Mx18
CB SRAM 2Mx72 (16MB)
1 BANKS 4-512kx36
FB SRAM 512kx36 (2MB)
CB SRAM 256kx128 (4MB)
LL SRAM 512kx36 (2MB)
PM
512
MID PLANE
64-500 MHz Tx PAIRS
64-500 MHz Rx PAIRS
UDASL BLOCK 1 X
CPU I/F BUS
4x UDASL
501
PM BLOCK 1 X

FIG. 4C

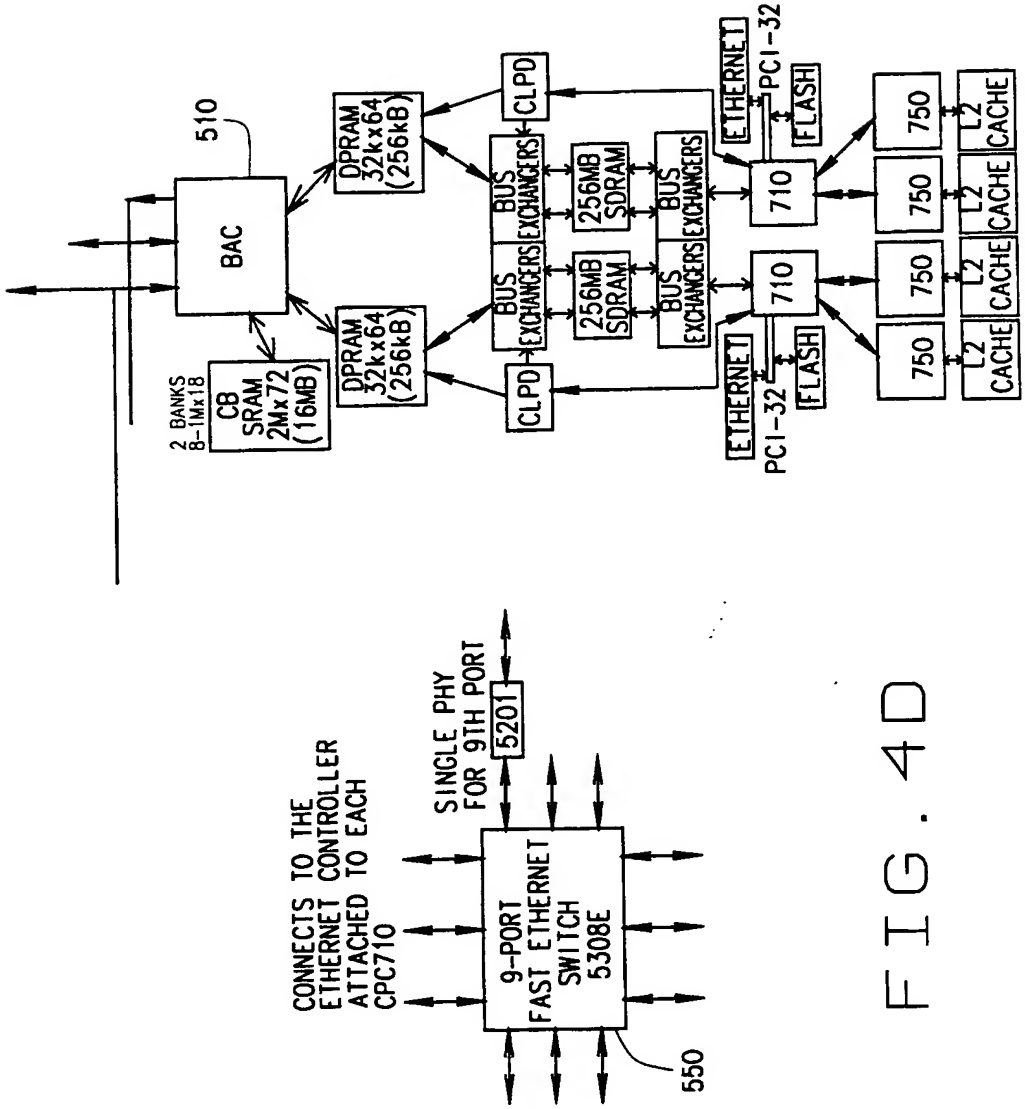

510
BAC
DPRAM 32kx64 (256kB)
DPRAM 32kx64 (256kB)
CB SRAM 2Mx72 (16MB)
2 BANKS 8-1Mx18
CLPD
CLPD
BUS EXCHANGERS
BUS EXCHANGERS
256MB SDRAM
256MB SDRAM
BUS EXCHANGERS
BUS EXCHANGERS
ETHERNET
PCI-32
FLASH
710
710
ETHERNET
PCI-32
FLASH
750
750
750
750
L2 CACHE
L2 CACHE
L2 CACHE
L2 CACHE
CONNECTS TO THE ETHERNET CONTROLLER ATTACHED TO EACH CPC710
SINGLE PHY FOR 9TH PORT
5201
9-PORT FAST ETHERNET SWITCH 5308E
550

FIG. 4D

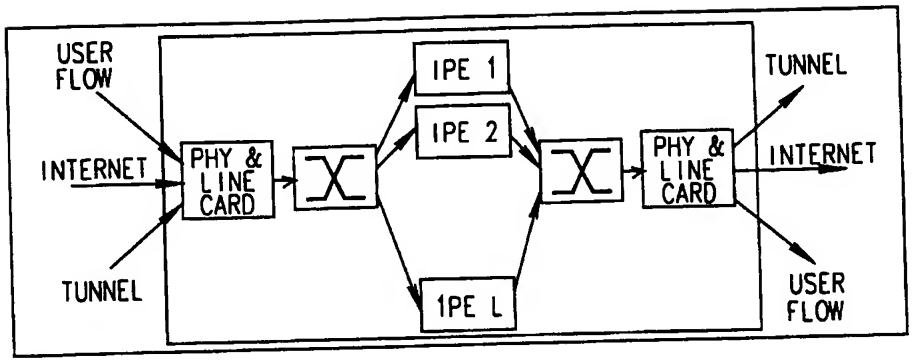

USER FLOW
INTERNET
TUNNEL
PHY & LINE CARD
IPE 1
IPE 2
IPE L
PHY & LINE CARD
TUNNEL
INTERNET
USER FLOW

FIG. 5

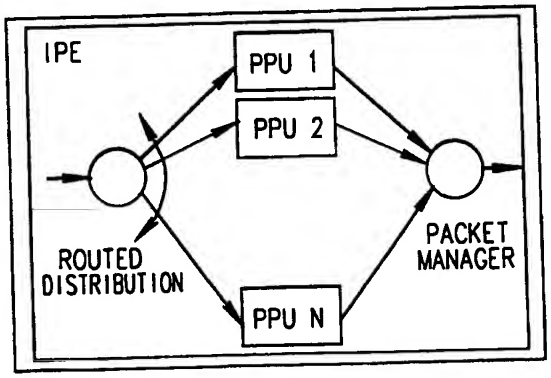

IPE
PPU 1
PPU 2
PPU N
ROUTED DISTRIBUTION
PACKET MANAGER

FIG. 6

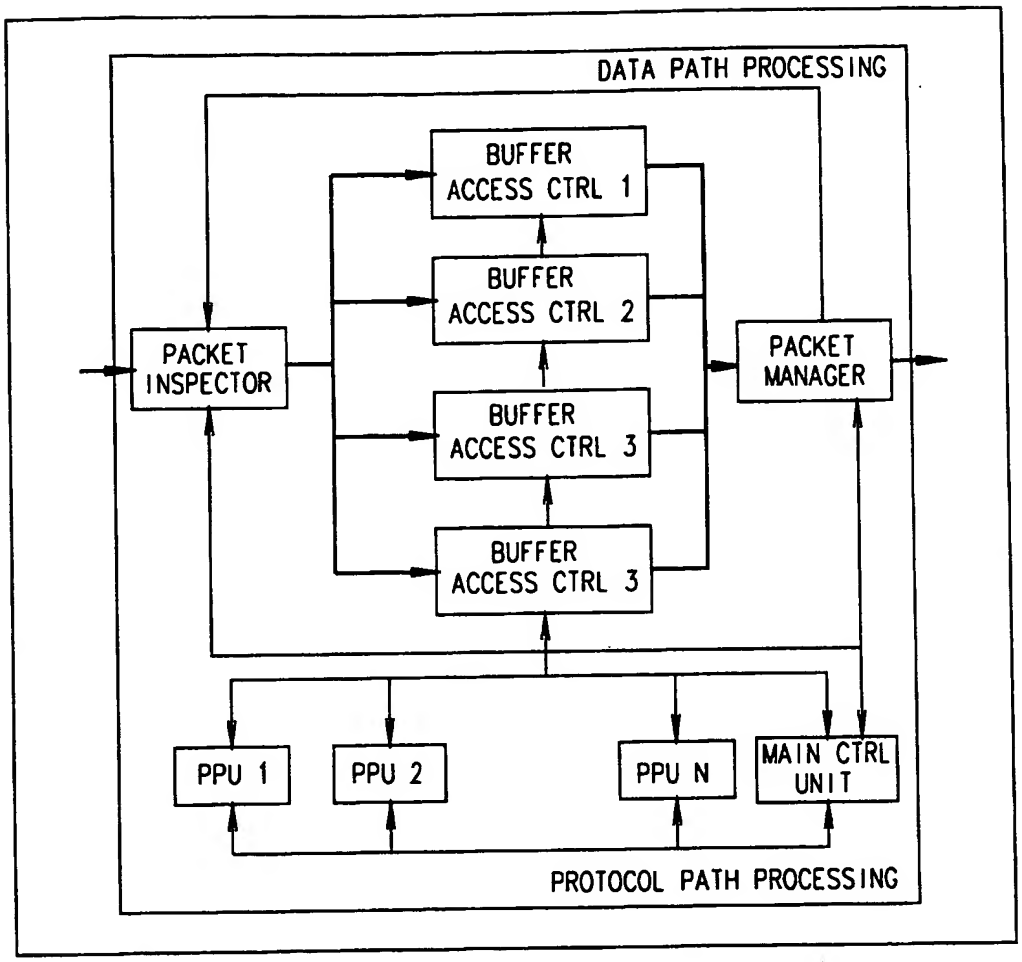

DATA PATH PROCESSING
BUFFER ACCESS CTRL 1
BUFFER ACCESS CTRL 2
BUFFER ACCESS CTRL 3
BUFFER ACCESS CTRL 3
PACKET INSPECTOR
PACKET MANAGER
PPU 1
PPU 2
PPU N
MAIN CTRL UNIT
PROTOCOL PATH PROCESSING

FIG. 7

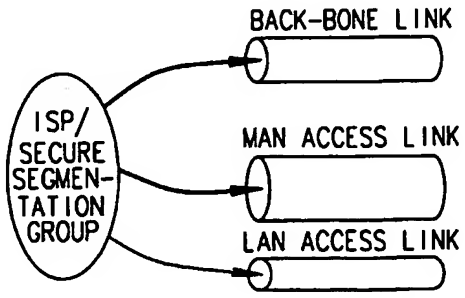

BACK-BONE LINK
ISP/ SECURE SEGMEN-TATION GROUP
MAN ACCESS LINK
LAN ACCESS LINK

FIG. 12

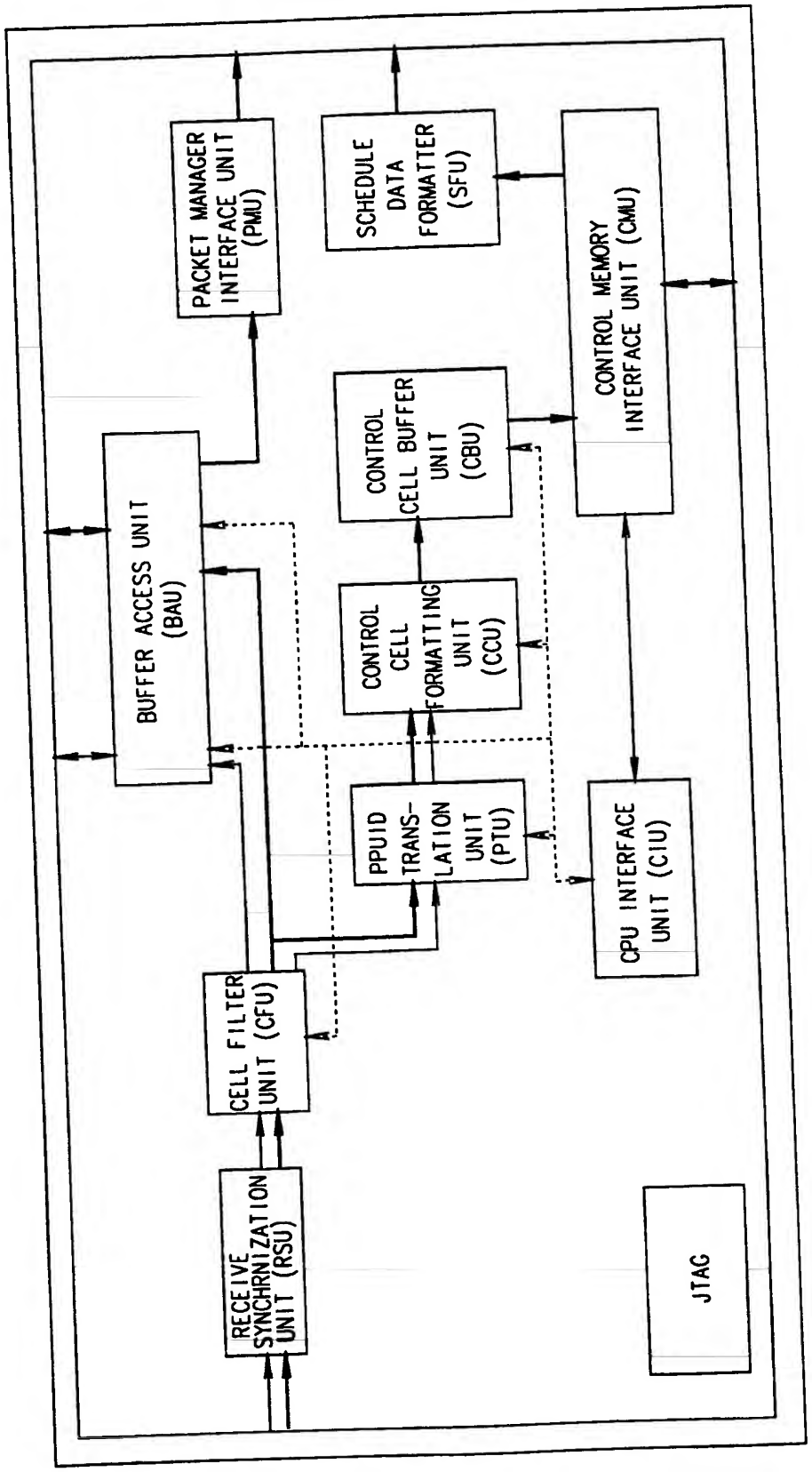

PACKET MANAGER INTERFACE UNIT (PMU)
SCHEDULE DATA FORMATTER (SFU)
CONTROL MEMORY INTERFACE UNIT (CMU)
CONTROL CELL BUFFER UNIT (CBU)
BUFFER ACCESS UNIT (BAU)
CONTROL CELL FORMATTING UNIT (CCU)
PPUID TRANS-LATION UNIT (PTU)
CPU INTERFACE UNIT (CIU)
CELL FILTER UNIT (CFU)
RECEIVE SYNCHRNIZATION UNIT (RSU)
JTAG

FIG. 8

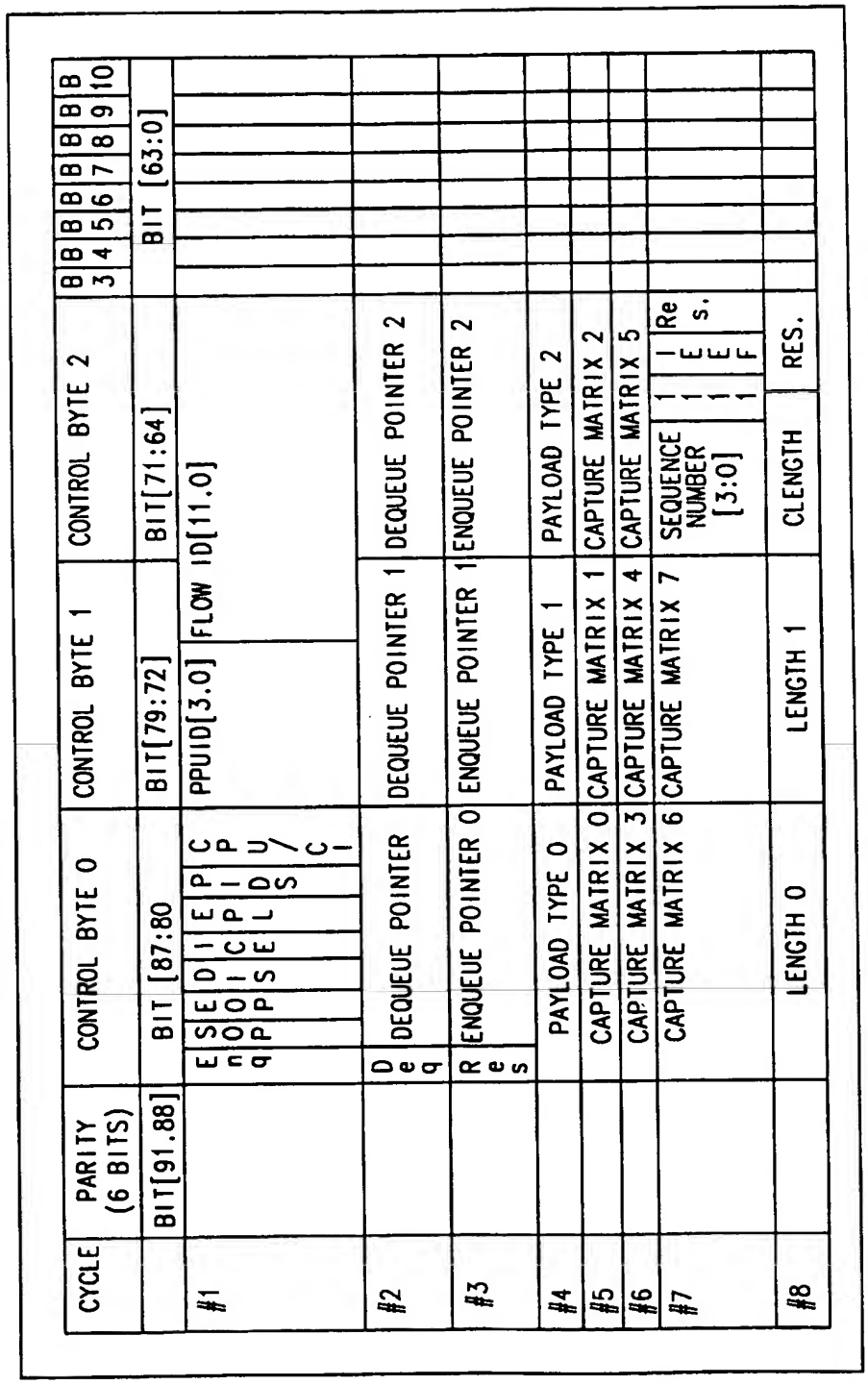

CYCLE
PARITY (6 BITS)
CONTROL BYTE 0
CONTROL BYTE 1
CONTROL BYTE 2
B 3
B 4
B 5
B 6
B 7
B 8
B 9
B 10
BIT[91.88]
BIT [87:80
BIT[79:72]
BIT[71:64]
BIT [63:0]
Enq
SEQ NO PPS
DISCARD SELECT
EPD
CP
PPUID[3.0]
FLOW ID[11.0]
#1
Deq
DEQUEUE POINTER
DEQUEUE POINTER 1
DEQUEUE POINTER 2
#2
Res
ENQUEUE POINTER 0
ENQUEUE POINTER 1
ENQUEUE POINTER 2
#3
PAYLOAD TYPE 0
PAYLOAD TYPE 1
PAYLOAD TYPE 2
#4
CAPTURE MATRIX 0
CAPTURE MATRIX 1
CAPTURE MATRIX 2
#5
CAPTURE MATRIX 3
CAPTURE MATRIX 4
CAPTURE MATRIX 5
#6
CAPTURE MATRIX 6
CAPTURE MATRIX 7
SEQUENCE NUMBER [3:0]
1 1 1 1
IEEF
Res.
#7
LENGTH 0
LENGTH 1
CLENGTH
RES.
#8

FIG. 9A

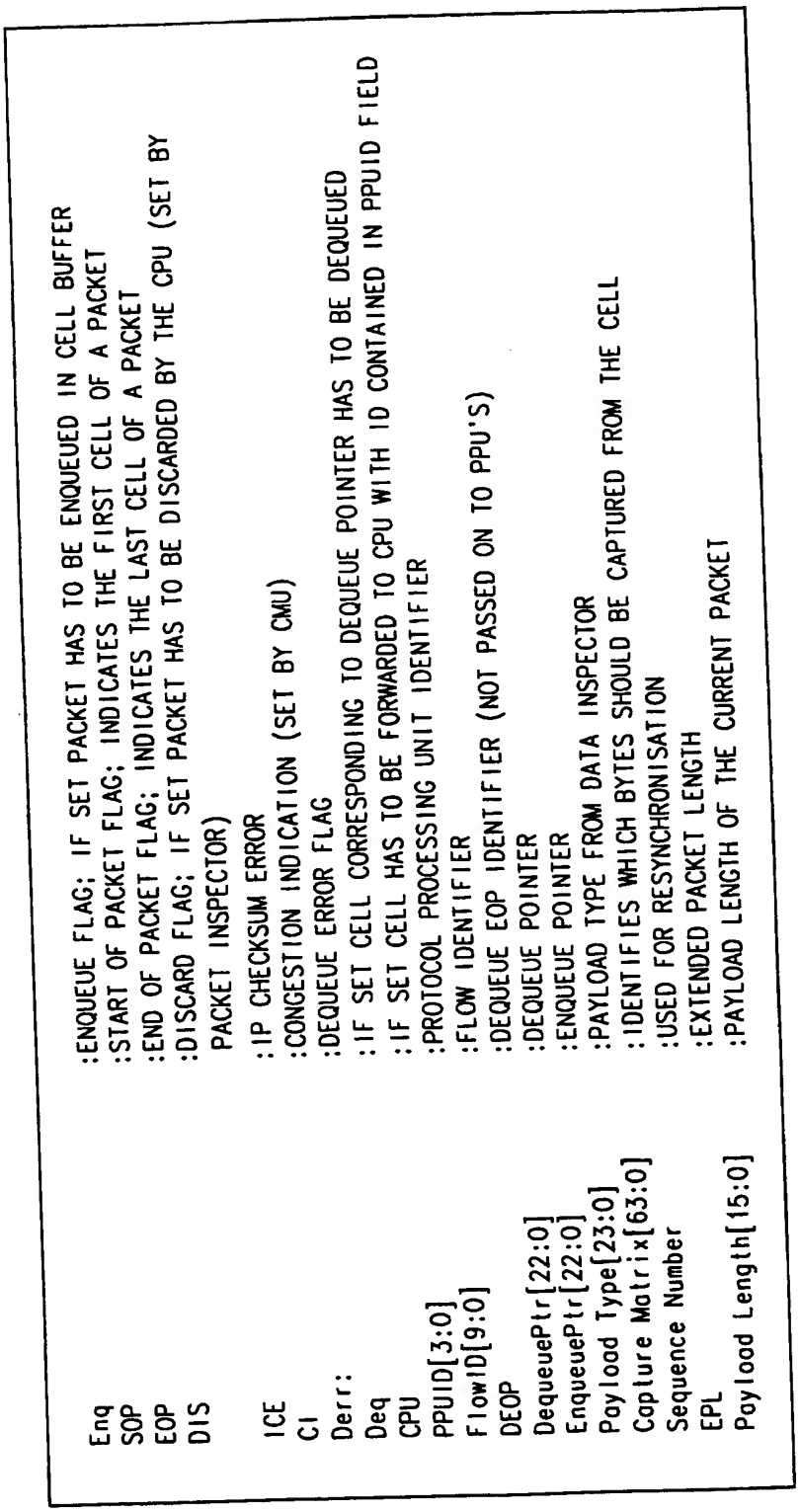

Enq :ENQUEUE FLAG; IF SET PACKET HAS TO BE ENQUEUED IN CELL BUFFER
SOP :START OF PACKET FLAG; INDICATES THE FIRST CELL OF A PACKET
EOP :END OF PACKET FLAG; INDICATES THE LAST CELL OF A PACKET
DIS :DISCARD FLAG; IF SET PACKET HAS TO BE DISCARDED BY THE CPU (SET BY PACKET INSPECTOR)
ICE :IP CHECKSUM ERROR
CI :CONGESTION INDICATION (SET BY CMU)
Derr: :DEQUEUE ERROR FLAG
Deq :IF SET CELL CORRESPONDING TO DEQUEUE POINTER HAS TO BE DEQUEUED
CPU :IF SET CELL HAS TO BE FORWARDED TO CPU WITH ID CONTAINED IN PPUID FIELD
PPUID[3:0] :PROTOCOL PROCESSING UNIT IDENTIFIER
FlowID[9:0] :FLOW IDENTIFIER
DEOP :DEQUEUE EOP IDENTIFIER (NOT PASSED ON TO PPU'S)
DequeuePtr[22:0] :DEQUEUE POINTER
EnqueuePtr[22:0] :ENQUEUE POINTER
Payload Type[23:0] :PAYLOAD TYPE FROM DATA INSPECTOR
Capture Matrix[63:0] :IDENTIFIES WHICH BYTES SHOULD BE CAPTURED FROM THE CELL
Sequence Number :USED FOR RESYNCHRONISATION
EPL :EXTENDED PACKET LENGTH
Payload Length[15:0] :PAYLOAD LENGTH OF THE CURRENT PACKET

FIG. 9B

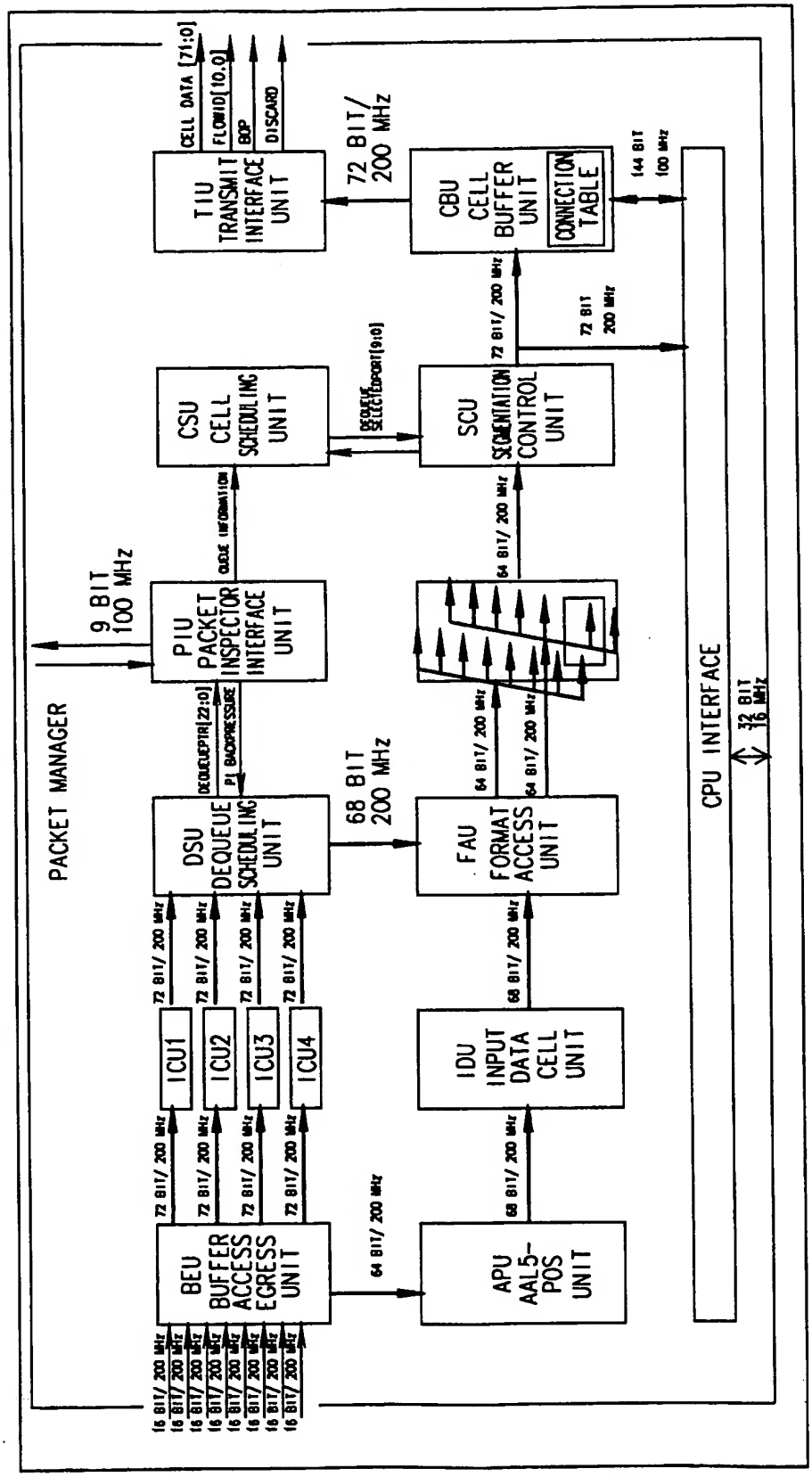

PACKET MANAGER
9 BIT 100 MHz
PIU PACKET INSPECTOR INTERFACE UNIT
TIU TRANSMIT INTERFACE UNIT
CELL DATA [71:0]
FLOWID[10:0]
EOP
DISCARD
72 BIT/ 200 MHz
CBU CELL BUFFER UNIT
CONNECTION TABLE
144 BIT 100 MHz
CSU CELL SCHEDULING UNIT
SCU SEGMENTATION CONTROL UNIT
72 BIT 200 MHz
64 BIT/ 200 MHz
DSU DEQUEUE SCHEDULING UNIT
68 BIT 200 MHz
FAU FORMAT ACCESS UNIT
CPU INTERFACE
32 BIT 16 MHz
ICU1
ICU2
ICU3
ICU4
IDU INPUT DATA CELL UNIT
BEU BUFFER ACCESS EGRESS UNIT
APU AAL5-POS UNIT

FIG. 10

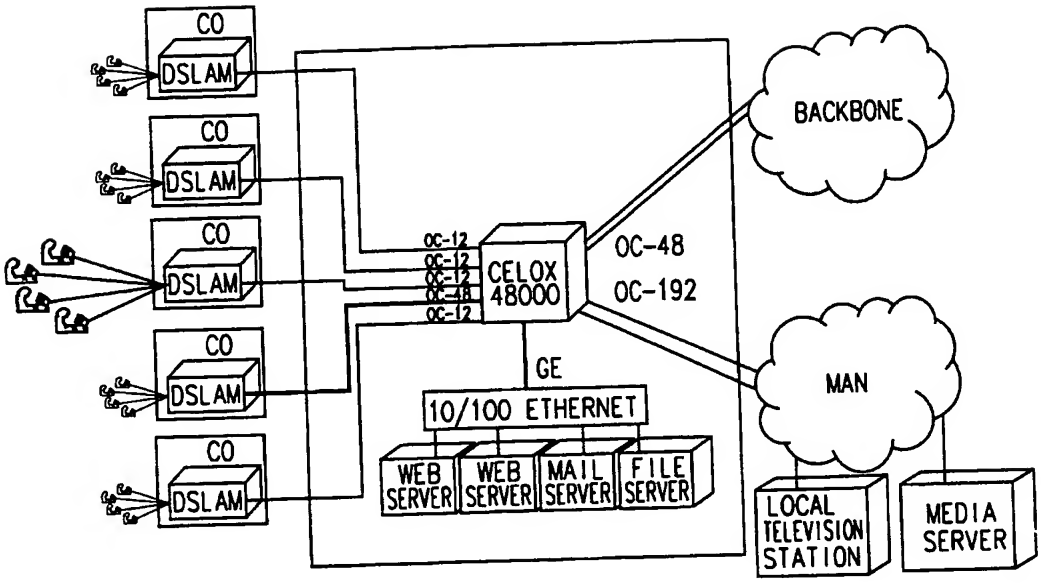

CO
DSLAM
CO
DSLAM
CO
DSLAM
CO
DSLAM
CO
DSLAM
OC-12
OC-12
OC-12
OC-48
OC-12
CELOX
48000
OC-48
OC-192
BACKBONE
GE
10/100 ETHERNET
WEB SERVER
WEB SERVER
MAIL SERVER
FILE SERVER
MAN
LOCAL TELEVISION STATION
MEDIA SERVER

FIG. 11

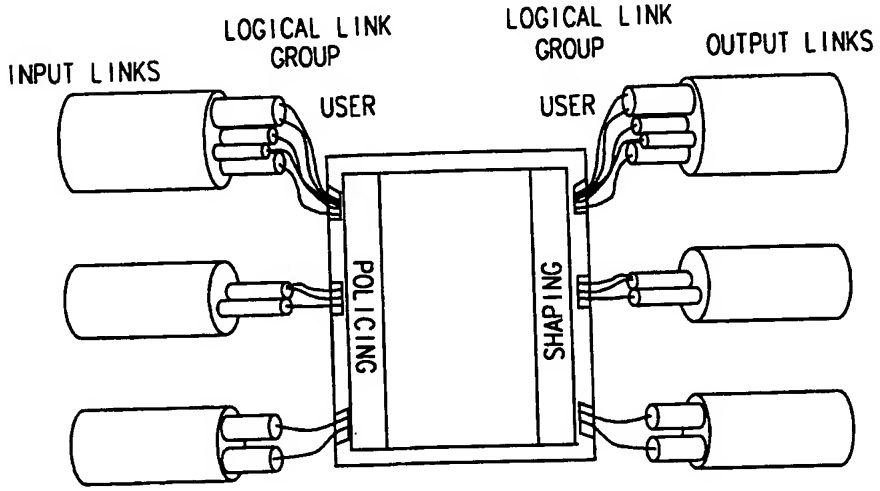

INPUT LINKS
LOGICAL LINK GROUP
USER
LOGICAL LINK GROUP
USER
OUTPUT LINKS
POLICING
SHAPING

FIG. 14

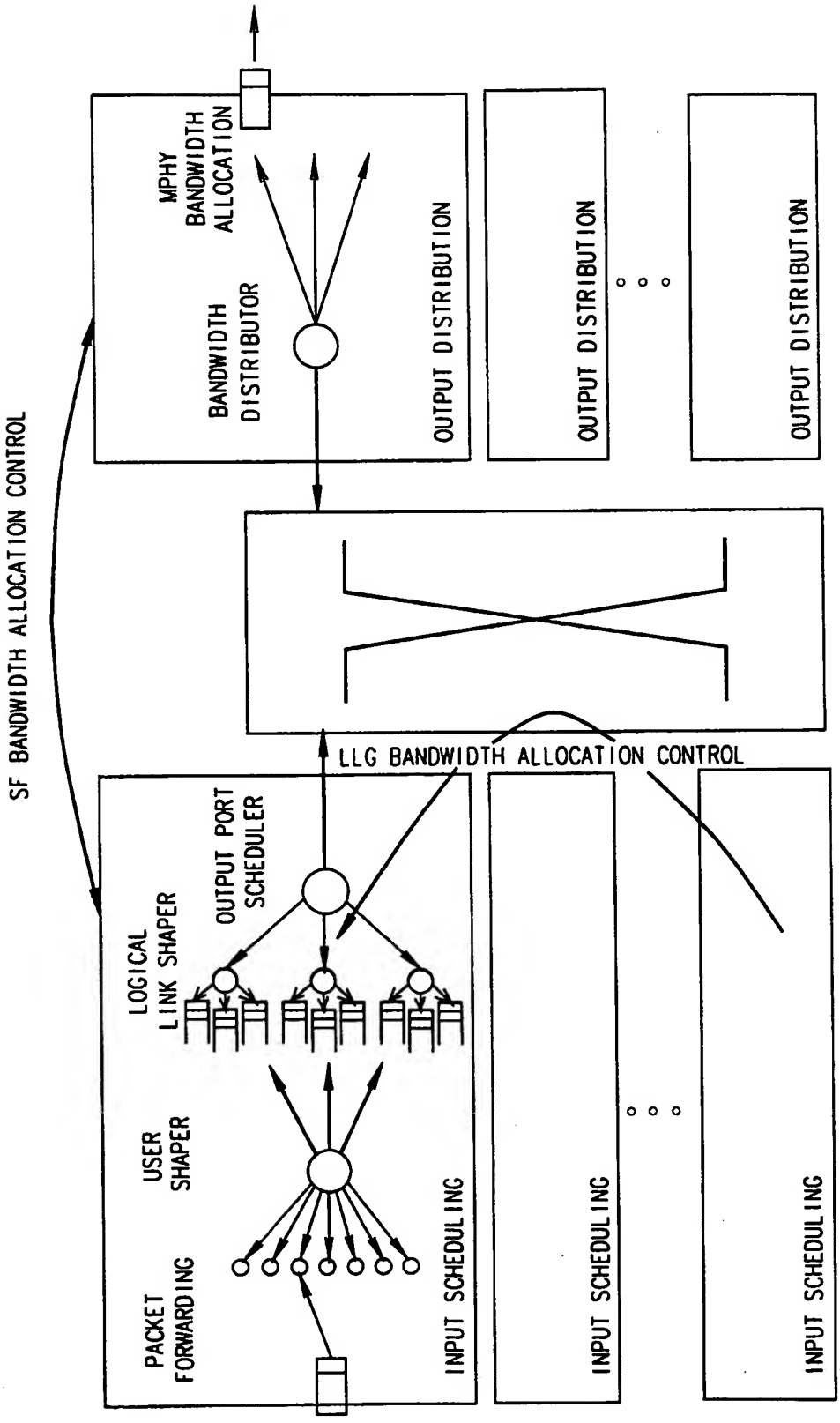

SF BANDWIDTH ALLOCATION CONTROL
MPHY BANDWIDTH ALLOCATION
BANDWIDTH DISTRIBUTOR
OUTPUT DISTRIBUTION
OUTPUT DISTRIBUTION
OUTPUT DISTRIBUTION
LLG BANDWIDTH ALLOCATION CONTROL
OUTPUT PORT SCHEDULER
LOGICAL LINK SHAPER
USER SHAPER
PACKET FORWARDING
INPUT SCHEDULING
INPUT SCHEDULING
INPUT SCHEDULING

FIG. 13

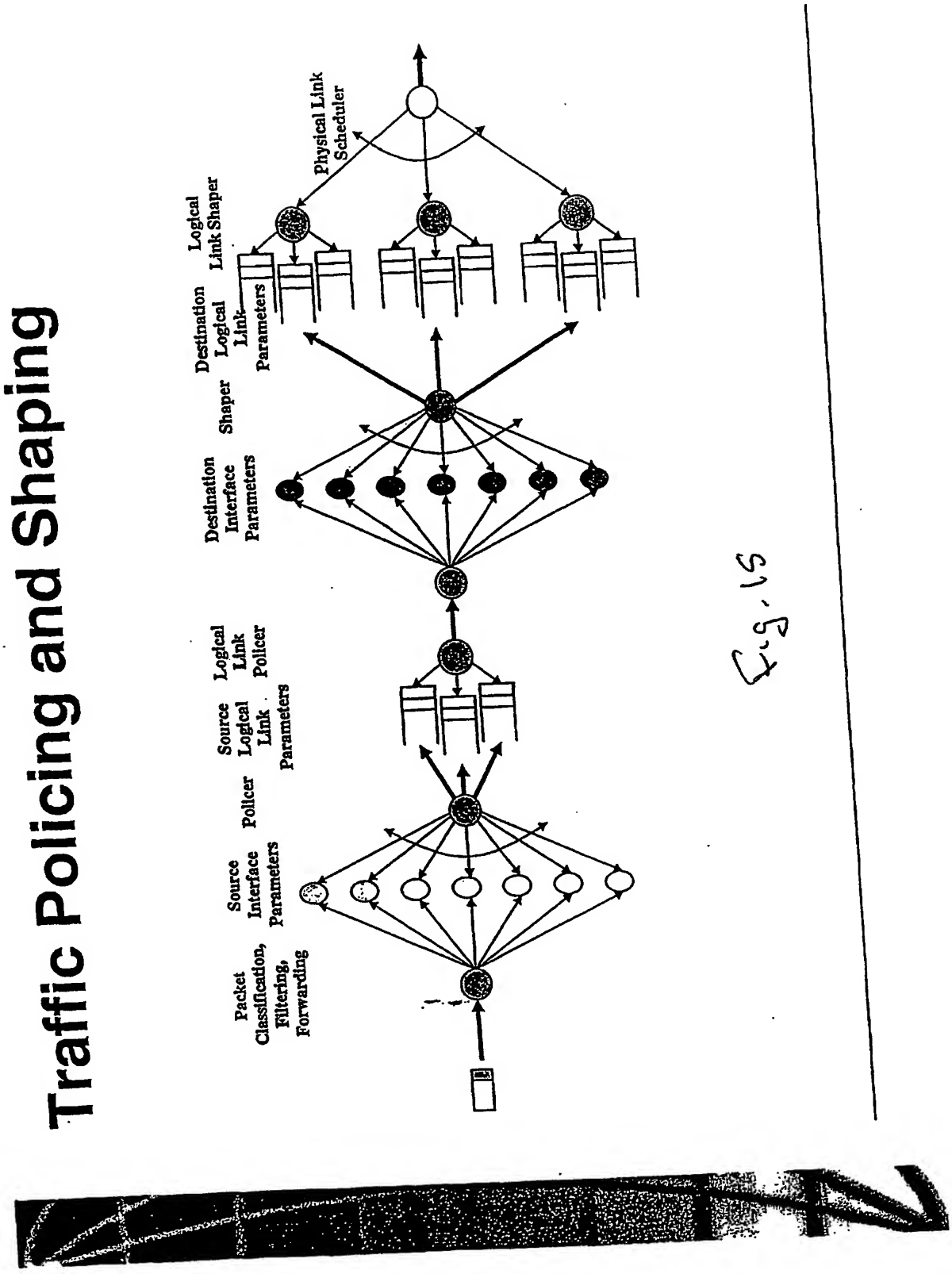

Traffic Policing and Shaping
Packet Classification, Filtering, Forwarding
Source Interface Parameters
Policer
Source Logical Link Parameters
Logical Link Policer
Destination Interface Parameters
Shaper
Destination Logical Link Parameters
Logical Link Shaper
Physical Link Scheduler

Fig. 15

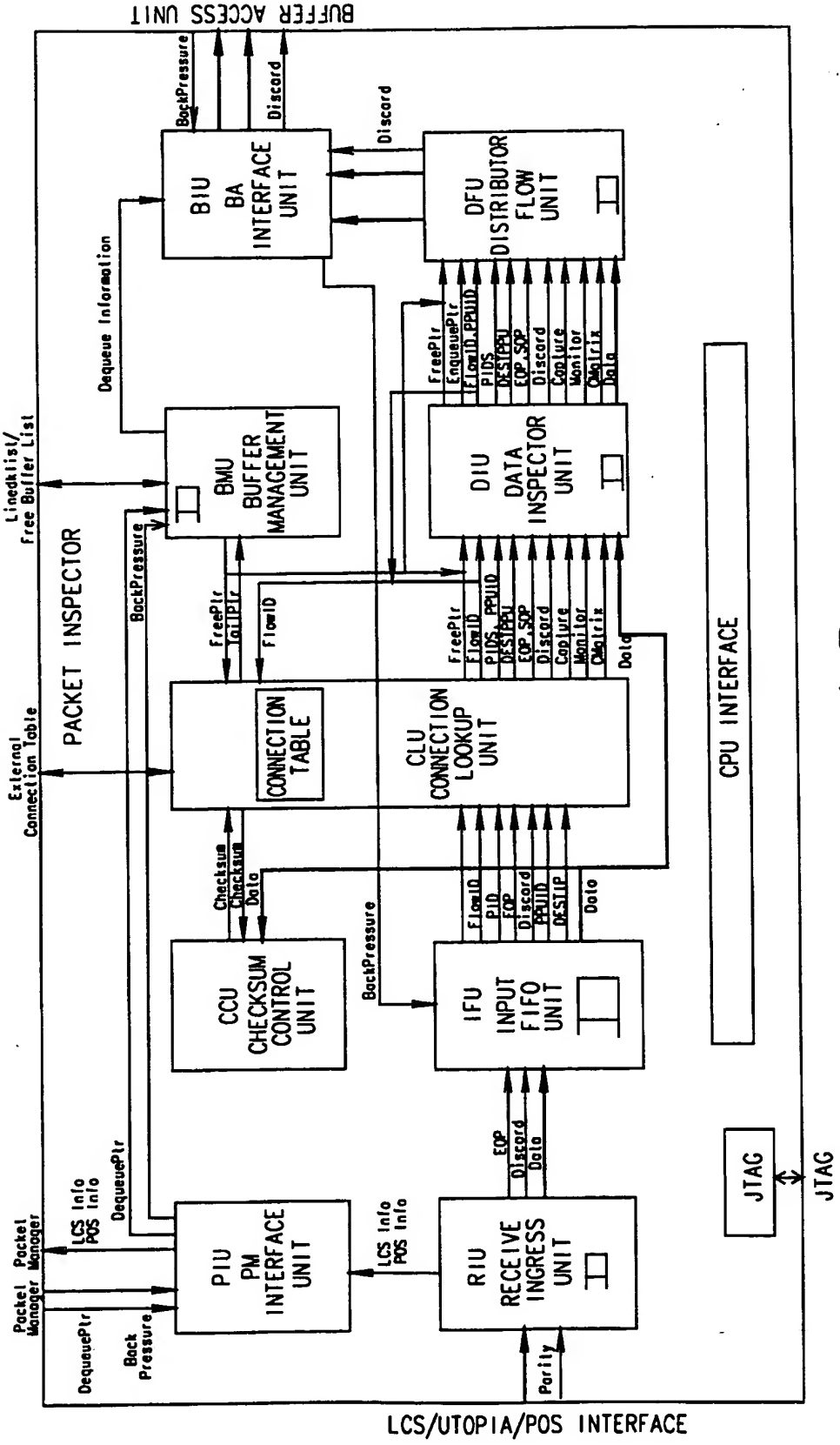

BUFFER ACCESS UNIT
BIU BA INTERFACE UNIT
DFU DISTRIBUTOR FLOW UNIT
BMU BUFFER MANAGEMENT UNIT
DIU DATA INSPECTOR UNIT
PACKET INSPECTOR
CONNECTION TABLE
CLU CONNECTION LOOKUP UNIT
CPU INTERFACE
CCU CHECKSUM CONTROL UNIT
IFU INPUT FIFO UNIT
PIU PM INTERFACE UNIT
RIU RECEIVE INGRESS UNIT
JTAG
LCS/UTOPIA/POS INTERFACE
BackPressure
Discard
Dequeue Information
FreePtr
EnqueuePtr
FlowID
PPUID
PIDS
DESTBU
EOP
SOP
Capture
Monitor
CMatrix
Data
LinedList/ Free Buffer List
External Connection Table
TailPtr
Checksum
PID
Parity
Packet Manager
LCS Info
POS Info
DequeuePtr
Back Pressure

FIG. 16

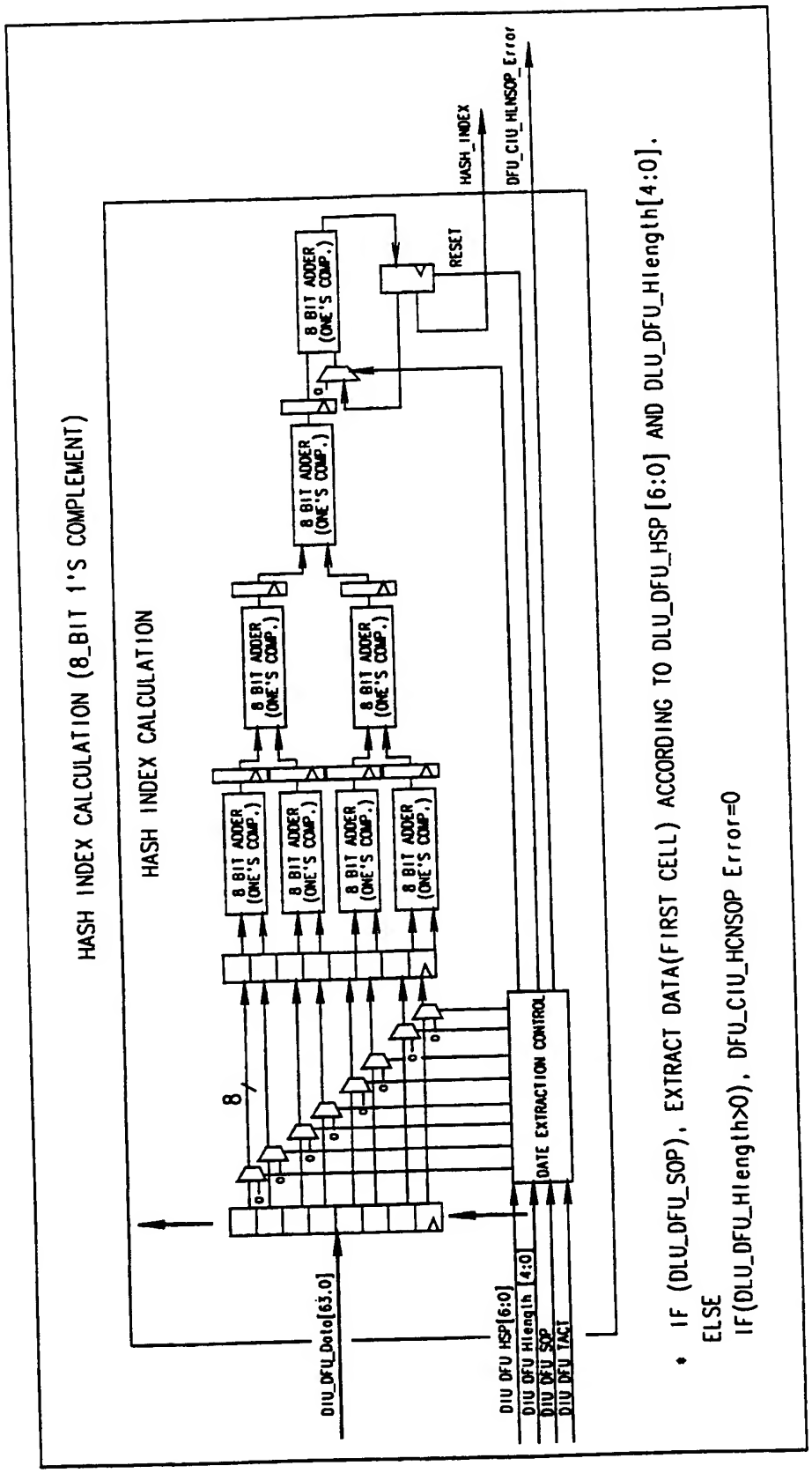

HASH INDEX CALCULATION (8_BIT 1'S COMPLEMENT)
HASH INDEX CALCULATION
8 BIT ADDER (ONE'S COMP.)
RESET
HASH_INDEX
DFU_CIU_HCNSOP_Error
DATE EXTRACTION CONTROL
DIU_DFU_Data[63:0]
DIU DFU HSP[6:0]
DIU DFU Hlength[4:0]
DIU DFU SOP
DIU DFU TACT
8
• IF (DLU_DFU_SOP), EXTRACT DATA(FIRST CELL) ACCORDING TO DLU_DFU_HSP[6:0] AND DLU_DFU_Hlength[4:0].
ELSE
IF(DLU_DFU_Hlength>0), DFU_CIU_HCNSOP Error=0

FIG. 17

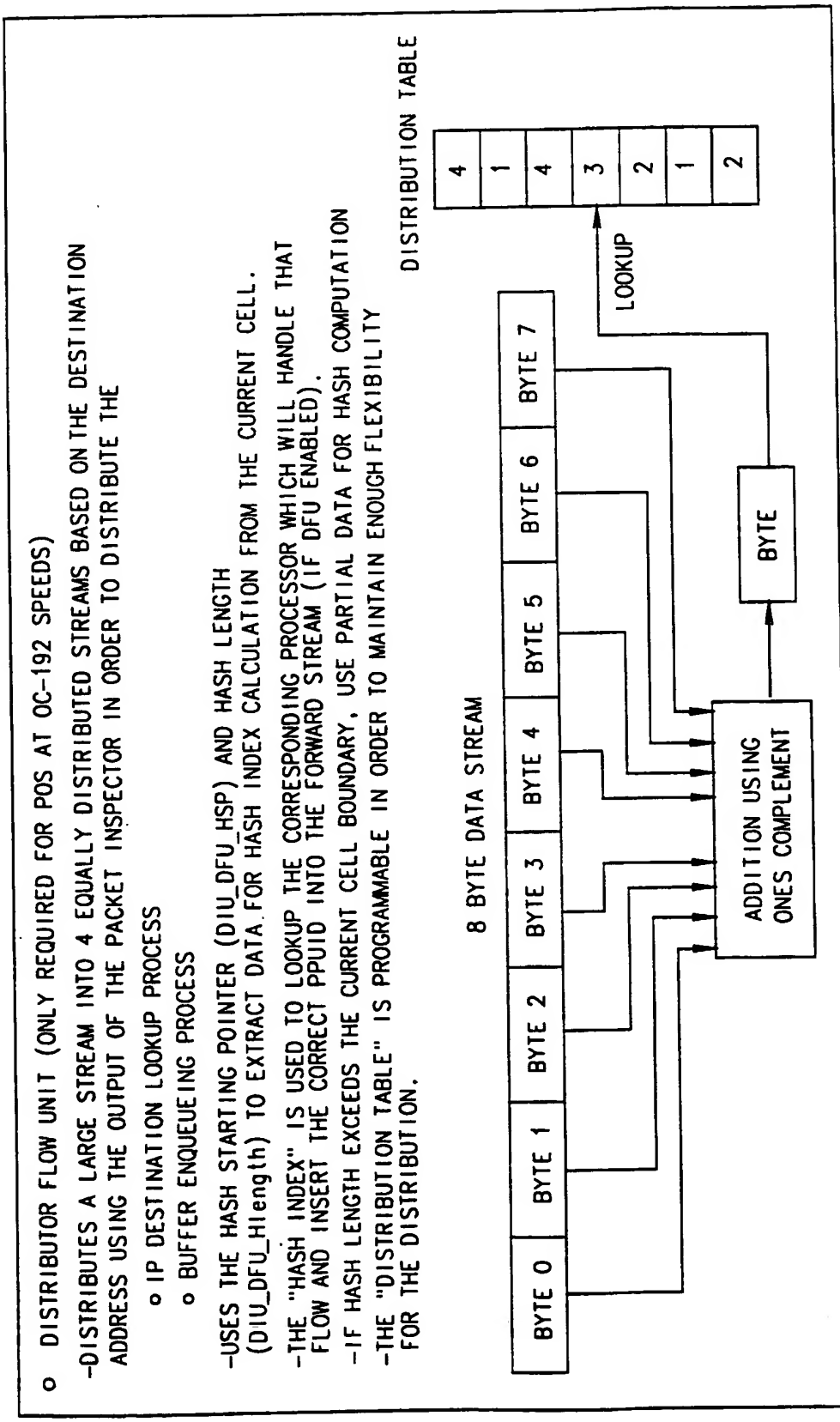

o DISTRIBUTOR FLOW UNIT (ONLY REQUIRED FOR POS AT OC-192 SPEEDS)
-DISTRIBUTES A LARGE STREAM INTO 4 EQUALLY DISTRIBUTED STREAMS BASED ON THE DESTINATION ADDRESS USING THE OUTPUT OF THE PACKET INSPECTOR IN ORDER TO DISTRIBUTE THE
o IP DESTINATION LOOKUP PROCESS
o BUFFER ENQUEUEING PROCESS
-USES THE HASH STARTING POINTER (DIU_DFU_HSP) AND HASH LENGTH (DIU_DFU_HIength) TO EXTRACT DATA FOR HASH INDEX CALCULATION FROM THE CURRENT CELL.
-THE "HASH INDEX" IS USED TO LOOKUP THE CORRESPONDING PROCESSOR WHICH WILL HANDLE THAT FLOW AND INSERT THE CORRECT PPUID INTO THE FORWARD STREAM (IF DFU ENABLED).
-IF HASH LENGTH EXCEEDS THE CURRENT CELL BOUNDARY, USE PARTIAL DATA FOR HASH COMPUTATION
-THE "DISTRIBUTION TABLE" IS PROGRAMMABLE IN ORDER TO MAINTAIN ENOUGH FLEXIBILITY FOR THE DISTRIBUTION.
DISTRIBUTION TABLE
4
1
4
3
2
1
2
LOOKUP
8 BYTE DATA STREAM
BYTE 0
BYTE 1
BYTE 2
BYTE 3
BYTE 4
BYTE 5
BYTE 6
BYTE 7
ADDITION USING ONES COMPLEMENT
BYTE

FIG. 18

INTERNATIONAL SEARCH REPORT

Inte[illegible]nal Application No
PCT/US 01/01003

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L12/56 H04Q11/04

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L H04Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 00 10297 A (TRIPATHI DEVENDRA K ;DEB ALAK K (US); SAMBAMURTHY NAMAKKAL S (US);) 24 February 2000 (2000-02-24)	1
A	page 11, line 3 -page 12, line 18 page 20, line 12 -page 21, line 20 page 23, line 7 -page 24, line 2	2-47
X	EP 0 944 288 A (NIPPON ELECTRIC CO) 22 September 1999 (1999-09-22)	12,21
Y	figures 3-6	32,47
A		1-11, 13-20, 22-31, 33-46
Y	EP 0 531 599 A (IBM) 17 March 1993 (1993-03-17)	32,47
A	column 4, line 11 -column 6, line 36	33-46

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
6 June 2001	22/06/2001

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax (+31-70) 340-3016	Meurisse, W

1

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

information on patent family members

Intl ial Application No
PCT/US 01/01003

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 0010297 A	24-02-2000	AU 5567499 A	06-03-2000
EP 0944288 A	22-09-1999	JP 11275114 A	08-10-1999
EP 0531599 A	17-03-1993	DE 69129851 D	27-08-1998
		DE 69129851 T	25-03-1999
		JP 2788577 B	20-08-1998
		JP 5219098 A	27-08-1993
		US 5311509 A	10-05-1994